Korespondenční Seminář z Programování

ZAČÁTEČNICKÁ KATEGORIE

29. ročník **KSP-Z** **Prosinec 2016**

Skončila druhá série KSP-Z a my vám přinášíme autorská řešení úloh. Věříme, že vám pomohou k tomu, abyste se v programování a hlavně v řešení problémů pořád zlepšovali. Gratulujeme všem, kdo získali nějaké body!

A jako obvykle se nás nebojte zeptat, pokud vám cokoliv není úplně jasné. Obrátit se na nás můžete přes fórum na našich stránkách nebo e-mailem na *ksp@mff.cuni.cz*.

Řešení druhé série začátečnické kategorie 29. ročníku KSP

29-Z2-1 Krocení zlé želvy

Naším úkolem bylo ze zadané posloupnosti příkazů zjistit, kde skončí Kevinova želva, až všechny tyto příkazy vykoná. Jako první je dobré si uvědomit, že úlohu můžeme řešit bez toho, abychom si vytvářeli velké pole a pohyb želvy po něm simulovali.

Tedy, abychom použili analogovou analogii, není potřeba vzít do ruky (velký) čtverečkovaný papír a postupně si kreslit kterými políčky želva projde. Namísto toho si stačí pamatovat vždy jen aktuální pozici a směr želvy.

Jakmile totiž zjistíme, jak ji posouvat, můžeme namísto simulace na velkém poli či po čtverečkovém papíře, prostě jen aplikovat změnu na zapamatovanou pozici či směr. V případě pokynu A konkrétně změnit pozici a při pokynech `<` a `>` otočení.

Pozice se skládá ze dvou souřadnic. Můžeme si představit třeba čtverečkovaný papír se čtverečkem $(0, 0)$ někde uprostřed.

Směr je ještě jednodušší, stačí si pamatovat, zda jsme otočení na sever, západ, jih nebo východ. Prakticky tedy postačí libovolná číselná proměnná. Na začátku je želva otočená nahoru.

Pro pořádky indexace si směry otočení seřadíme jak jsou napsané výše. Otočení s indexem 0 (přeci jen většina programovacích jazyků indexuje od 0, tak buďme konzistentní) bude na sever, s indexem 1 na západ a dále logicky proti směru hodinových ručiček.

Jakmile si vytvoříme takovouto reprezentaci, můžeme si všimnout, že otočení doleva je vždy zvýšení čísla otočení a doprava snížení. Problém jsou jen krajní hodnoty – když máme směr s indexem 3 (tj. na východ) a otočíme se doleva, dostaneme se na otočení na sever, tedy index 0.

Naštěstí ale nemusíme tyto krajní případy řešit separátně. Stačí místo upraveného indexu otočení vzít jeho zbytek po dělení čtyřmi, tj. např. $((i+1) \bmod 4)$, a dostaneme přesně to, co chceme.

Otáčení máme vyřešeno. Stačí už jen vymyslet, jak aktuální směr aplikovat v případě příkazu `A`, tedy posunu želvy dopředu. Nejjednodušší řešení je připravit si dvě pole o čtyřech prvcích. Jedno pro změnu souřadnice v x-ové a druhé v y-ové ose pro každý možný směr otočení.

Pro osu x může dané pole vypadat třeba takto: $[0, -1, 0, 1]$. Při otočení nahoru se při pohybu vpřed naše x-ová souřadnice nezmění. Při otočení doleva se zmenší o jedna a obdobně dál.

Při provádění příkazu `A` se pak jen podíváme do obou polí na hodnotu indexovanou otočením a danou hodnotu přičteme k odpovídající souřadnici aktuální pozice.

Časová složitost algoritmu je lineární s počtem příkazů, na každém totiž strávíme konstantně času. Paměťová je konstantní, pokud nepočítáme, že si je třeba pamatovat vstup. V průběhu algoritmu si totiž pamatujeme nezávisle na velikosti vstupu jen dvě pomocné proměnné.

Program (Python 3):
`http://ksp.mff.cuni.cz/viz/29-Z2-1.py`

Petr Houška

29-Z2-2 Sářina volba

Abychom se dozvěděli, kdo rozhodne o plánech na víkend, stačí spočítat počet Sářiných a Kevinových vítězství. Na vstupu dostaneme řetězce Sářiných a poté Kevinových tahů. Načteme si je odděleně do dvou proměnných. Zároveň si budeme pamatovat aktuální počet vítězství každého.

Pak již jen stačí procházet jednotlivé hry. Nejdříve vyhodnotíme první hru (tj. na indexu 0 v obou řetězcích), pak hru na indexu 1 až k poslední $n - 1$. K tomu nám poslouží například cyklus `for`, v jehož každém průchodu vítězi přičteme jedno vítězství.

Při procházení potřebujeme rozhodnout, kdo danou hru vyhrál. Uvědomíme si, že hra může skončit pouze devíti stavy (každý má tři možnosti jak zahrát), které můžeme snadno otestovat podmínkami.

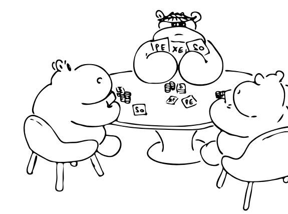
Pokud oba zahráli stejně, nevyhrál nikdo a přeskočíme rovnou na další hru.

Pokud Sára hrála kámen a Kevin nůžky, pak vyhrála Sára a připočteme ji jedno vítězství. Stejně tak v případě Sára nůžky + Kevin papír a Sára papír + Kevin kámen.

Pokud nenastala ani jedna z předchozích kombinací, pak nutně vyhrál Kevin, a proto mu připočteme jedno vítězství.

Program (Python 3):
`http://ksp.mff.cuni.cz/viz/29-Z2-2.py`

Jan Knížek



29-Z2-3 Petr v říši divů

V této úloze se ptáme, do kolika dalších políček lze v mřížce docestovat, pokud vyjdeme z jednoho konkrétního – Petrovy pozice. Na to se nejlépe hodí nějaké prohledávání. Chtěli bychom prohledávat od Petrovy pozice a každé políčko, na které dojdeme, si započítat a označit (nechceme některá políčka započítat vícekrát).

Prohledávat budeme následovně: vytvoříme si frontu, ve které budeme skladovat všechna políčka, která ještě chceme projít. Na začátku do ní přidáme jen políčko, na kterém stojí Petr. Pokaždé z fronty vytáhneme prvního kandidáta a podíváme se na jeho sousední políčka. Do fronty potom přidáme všechna políčka, která jsme ještě nenavštívili a nejsou to zdi.

Sousední políčka poznáme tak, že se jejich souřadnice liší o ± 1 od aktuálního políčka (například políčko napravo má souřadnici x větší o 1, a y stejnou). Každé políčko, které jsme do fronty přidali, započteme, označíme a potom vytáhneme další.

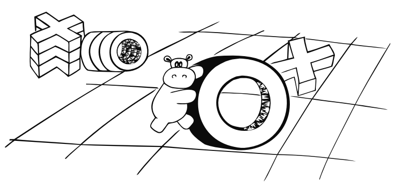
Jakmile bude fronta prázdná, tak jsme dokončili procházení z Petrovy pozice a nikam jinam se už nedostaneme. Zbývá tedy vypsat číslo, které jsme napočítali a skončit.

Poslední malý problém k vyřešení je, jak najít pozici, na které se nachází Petr a Sára. Protože mohou být kdekoliv v mapě, nepomůže nám žádný chytrý algoritmus a musíme zkontrolovat všechna políčka. Budeme procházet od začátku jedno po druhém, dokud nenarazíme na znak `P` a jeho pozici si zapamatujeme.

Zbývá si rozmyslet, zda se nemůže stát, že v lese existuje políčko, na které by se Petr a Sára mohli dostat, ale my jsme ho nezapočítali. To se nestalo, protože do fronty se během prohledávání dostalo každé dostupné políčko. Naopak, nezapočetli jsme náhodou nějaká navíc? Protože jsme přičítali pouze volná políčka, na která jsme se mohli dostat, nezapočetli jsme nic navíc.

Jak je to celé rychlé a kolik to žere paměti v počítači? Nalezení Petra se Sárou trvá $\mathcal{O}(R \cdot S)$, prohledávání trvalo také $\mathcal{O}(R \cdot S)$, protože každé políčko se do fronty dostalo maximálně jedenkrát. Celý program je tedy lineární a dokonce trvá pouze $\mathcal{O}(R \cdot S + R \cdot S) = \mathcal{O}(R \cdot S)$

Paměťová složitost je také lineární $\mathcal{O}(R \cdot S)$, protože jediné, co si potřebujeme pamatovat je mapa, která má $R \cdot S$ políček, a fronta, ve které se nikdy více políček než jich je v mapě neobjeví. Pokud si nejsi jistý v tom, co v tomto odstavci řešíme, můžeš si přečíst kuchařku o složitosti.[1]



Na závěr podotkněme, že program se určitě zastaví, protože hledání Petra skončí po maximálně N krocích a naše prohledávání se zastaví, neboť každé políčko dáme do fronty maximálně jednou.

Jako třešničku na dortu přidáme, že algoritmus, který jsme použili při řešení je ve světě informatiky tak rozšířený, že má své vlastní jméno. Jmenuje se BFS – prohledávání do šířky a dá se použít na různé grafové problémy. Pokud nevíš, co graf je, mohla by tě zajímat naše kuchařka o grafech.[2] Pokud už grafy znáš, tak si můžeš rozmyslet, že čtverečková síť, jako naše mapa ze zadání, je vlastně graf. Stačí každé políčko prohlásit za vrchol a spojit ho čtyřmi hranami s jeho sousedy.

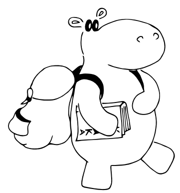
Program (Python 3):
`http://ksp.mff.cuni.cz/viz/29-Z2-3.py`

Štěpán Hojdar

29-Z2-4 Zuzka: Cesta tam a zase zpátky

Zadání po nás chtělo, abychom Zuzce nalezli nejdelší úsek, ve kterém půjdeme nejvýše K sekund do kopce. Nejprve vyzkoušíme jednoduché řešení.

Pro každé místo v posloupnosti vyzkoušíme, jestli náhodou ta nejdelší podposloupnost nezačíná zrovna tam. Tedy si naprogramujeme funkci, které řekneme začátek, a ona od něj najde nejdelší možný úsek, ve kterém půjde maximálně K sekund do kopce.



To je přeci jednoduché – pořídíme si energetickou kasičku, do které vložíme K penízků, a postavíme Kevina na začátek. Poté necháme Kevina jít co nejdále to půjde. Z kopce a po rovině půjde zadarmo, ale do kopce bude chtít penízek z kasičky. Jakmile bude chtít Kevin penízek, ale kasička bude prázdná, skončíme. Stejně tak pokud dojde Kevin na úplný konec trasy.

Z takto nalezených posloupností už jen snadno vybereme tu nejdelší. Jak to celé bude rychlé? Představte si dlouhou cestu, která bude ovšem celá z kopce. Naše funkce tedy pokaždé dojde až na konec cesty a celkem vykoná $N + (N-1) + \ldots + 1$ kroků, což je ovšem součet aritmetické posloupnosti, který vyjde v $\mathcal{O}(N^2)$.

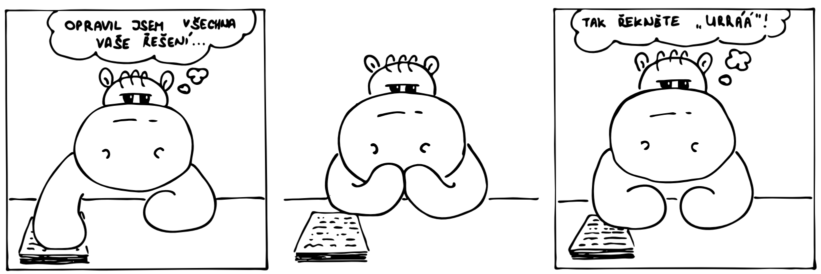
Chudák Kevin musí totiž pořád chodit tu samou trasu znovu a znovu. Proveďme jednoduché pozorování: pokud posuneme začátek doprava, Kevin může dojít jen dál – konec se nemůže posunout vlevo. Kevin se tedy nemusí vůbec vracet!

Na začátku algoritmu postavíme jak Zuzku tak Kevina na začátek trasy. Kevina pošleme kupředu, aby došel co nejdále, ale maximálně K sekund šel do kopce. Od této chvíle bude platit následující *invariant* (tvrzení, jehož platnost se v průběhu algoritmu nemění): mezi Kevinem a Zuzkou bude právě K sekund chůze do kopce.

Teď přijde řada na Zuzka. Ta půjde dopředu, ale pouze z kopce či po rovině. Kdykoli by měla jít do kopce, zavolá Kevinovi, a sekundu do kopce půjdou společně. Poté si Zuzka může odpočinout a Kevin utéct Zuzce co nejdále z kopce.

[1] `http://ksp.mff.cuni.cz/viz/kucharky/slozitost`
[2] `http://ksp.mff.cuni.cz/viz/kucharky/grafy`

	řešitel	*škola*	*ročník*	*sérií*	*Z2-1*	*Z2-2*	*Z2-3*	*Z2-4*	*Z2-5*	*Z2-6*	*série*	*celkem*
104.–105.	Evgeniya Knyazeva	GNVPlániPH	3	2		10					10,0	24,0
	Roman Ondráček	GBoskovice	3	7							0,0	24,0
106.	Tomáš Dostál	MendelG_OP	2	1	8	10	1	4			23,0	23,0
107.–108.	Lukáš Vavřík	GNeumannŽR	3	1							0,0	22,0
	Marek Zelený	GVoděraPH	3	1							0,0	22,0
109.	Tuan Anh Hoang	GZborovPH	3	3	8	10	0				18,0	21,0
110.–111.	Ondřej Buček	GJarošeBO	3	1	8	8		4			20,0	20,0
	Ondřej Gajda	GTěš	0	2					9		9,0	20,0
112.–123.	Dávid Daubner	GVaršŽilina	2	1							0,0	18,0
	Dominik Dinh	GNVPlániPH	2	1	8	10					18,0	18,0
	Vít Gaďurek		2	6							0,0	18,0
	Alexandra Géciová	GJHroncaBA	1	2	8	10	0	0			18,0	18,0
	Tereza Hladíková	JazG HK	3	1							0,0	18,0
	Štěpán Košan	GKlatovy	4	2							0,0	18,0
	Jiří Kvapil	GTomkovaOL	−1	2	8	10	0				18,0	18,0
	Václav Luňák	GDašickáPA	4	1	8	10					18,0	18,0
	Ondřej Mašek	GEBenešeKL	3	1							0,0	18,0
	Tomáš Novotný	G BO-Řeč	3	2	8	10					18,0	18,0
	Pavel Turinský	G Brandýs	4	5							0,0	18,0
	Matěj Šmíd	SPŠÚžlabPH	3	4	8	10					18,0	18,0
124.–126.	Martin Hubata	GMikulášPL	1	3	6	1		2			9,0	16,0
	Arian Adam Ott	GSOŠRok	0	2	8						8,0	16,0
	David Rajchman	MasG_Plzeň	0	2	8						8,0	16,0
127.	Petr Doubravský	AkademG_PH	1	2	2	10					12,0	15,0
128.	Tomáš Chabada	SPŠMasarLI	1	1	6	8					14,0	14,0
129.	Jan Koška	GJírovcČB	−3	1		10	3				13,0	13,0
130.–131.	Ondřej Hráček	GOlgHavl	0	1							0,0	12,0
	Vojtěch Lengál	GZborovPH	3	1							0,0	12,0
132.	Mária Ďuračková	GJHroncaBA	2	1							0,0	10,0
133.–138.	Matúš Ferech	GJHroncaBA	4	1							0,0	8,0
	Frantisek Hanzlik	ZŠ Elem	−1	1							0,0	8,0
	Jan Hřebenář	20. ZŠ	0	1							0,0	8,0
	Jan Kutálek	GSOŠNovJicin	3	1							0,0	8,0
	Šimon Prokop	21. ZŠ	0	1	8				0		8,0	8,0
	Natálie Volfová	GJírovcČB	1	1							0,0	8,0
139.–140.	Petr Chotěborský	GSla	0	1							0,0	6,0
	Jakub Švojgr	GČeskáČB	−2	1	6						6,0	6,0
141.	Petr Kabourek	G BO-Řeč	1	1	4	0					4,0	4,0
142.	Tomáš Husák	GLitoměřPH	3	1							0,0	3,0
143.	Filip Bouda	SŠStrŠvejc	1	1							0,0	2,0
144.	Štěpán Henrych	GŽat	1	1		0,3					0,3	0,3



KSP pro vás připravují studenti Matematicko-fyzikální fakulty Univerzity Karlovy.

Webové stránky: *https://ksp.mff.cuni.cz/* **E-mail:** *ksp@mff.cuni.cz* **Diskusní fórum:** *https://ksp.mff.cuni.cz/forum/*

Chcete-li s námi komunikovat bezpečně, můžete si ověřit náš HTTPS certifikát – jeho SHA1 fingerprint je: `E9:DB:EE:C6:62:BC:14:DE:09:E4:E8:97:DC:36:0E:87:B3:50:B0:01`.

Zajímá nás chvíle, kdy budou Kevin se Zuzkou nejdále od sebe. To je totiž díky invariantu chvíle, kdy Kevin se Zuzkou vymezují nejdelší podposloupnost podle zadání.

Jakmile Kevin dojde na konec trasy, můžeme rovnou skončit, protože Zuzka už se bude jen přibližovat.

Teď už víme, že algoritmem vydané řešení bude vždy správné, ale ještě nám zbývá ukázat, že pokud řešení existuje, algoritmus ho najde. To je naštěstí jednoduché, protože každé řešení musí někde začínat a přes to místo musí Zuzka někdy přejít.

Algoritmicky samozřejmě nebudeme posílat Zuzce a Kevinovi itinerář cesty, ale budeme si posouvat dva ukazatele nad polem. Když si uvědomíte, že oba ukazatele posouváte pouze vpravo, vyjde z toho optimální časová složitost $\mathcal{O}(N)$.

Paměťovou složitost máme ovšem zatím také lineární. Pokud bychom chtěli ušetřit, musíme se vydat ještě o krůček dál. Než si ale přečtete další odstavce, načíst celý soubor se vstupem do paměti k řešení většinou bohatě stačí. Následující informace tedy berte jako teoretický bonus.

Všimněte si, že v celém algoritmu nás vlastně vůbec nezajímaly konkrétní hodnoty nadmořských výšek. Místo nich nám stačí uvažovat, kdy cesta vedla do kopce a kdy z kopce. Dokonce ani nepotřebujeme mít jednotlivé sekundy v paměti rozdělené – stačí nám koukat na to, jak dlouhé jsou úseky z kopce mezi jednotlivými sekundami do kopce.

Mohli bychom tedy algoritmus pozměnit tak, že by načítal jednotlivá čísla s tím, jak přesouvá Kevina, a pro Zuzku už by si pamatoval jen délku dalších K úseků z kopce. Tím bychom srazili paměťovou složitost na $\mathcal{O}(K)$.

Program (Python 3):
`http://ksp.mff.cuni.cz/viz/29-Z2-4.py`

Ondra Hlavatý

29-Z2-5 Dva roky bez prázdnin

Od Šklíby jsme dostali úkol najít původce řetězu putování spamu mezi bytostmi. Nejprve si můžeme rozmyslet, že řetěz rozesílání e-mailu nám tvoří nějaký orientovaný graf. Seznam dvojic, jenž Šklíba získal, není nic jiného, než seznam hran, avšak neorientovaných, v tomto grafu. Vrcholy potom reprezentují jednotlivé bytosti, které se dostaly do styku se spamem.

Dále víme, že každé bytosti spam přišel nejvýše jednou a možná jej odeslala dalším K bytostem. V řeči grafů to znamená, že nejvýše jedna hrana do vrcholu (bytost) vstupuje a vystupuje z něj buď K hran, nebo žádná.

Stupněm vrcholu budeme značit počet hran, které vedou do nebo z daného vrcholu. Podívejme se tedy na možnosti, jaký může být stupeň vrcholu bytosti:

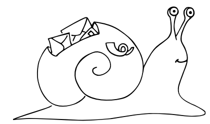
- Pokud bytosti spam přišel, do jejího vrcholu vede právě jedna hrana, stupeň se tedy zvýší o 1.
- Pokud bytost spam odeslala dalším K bytostem, stupeň jejího vrcholu se zvýší o K.

Pojďme z toho prozkoumat, jakého tvaru tento graf nabývá. Již víme, že každý vrchol má nejvýše jednoho předchůdce. Také má každý vrchol právě 0 nebo K potomků. Takový graf nápadně připomíná strom.

Kdybychom se nezabývali orientací hran, opravdu o strom jde. Kořenem je v tomto případě původce spamu a listy jsou doručitelé, kteří dále spam nerozeslali. Dále si můžeme všimnout, že je tento strom k-ární, jelikož každý vrchol, jenž není list, má právě k potomků.

Stupně vrcholů v našem grafu tedy můžou být 1 pro „listy“, K pro „kořen“, nebo $K+1$ pro ostatní. Z toho můžeme usoudit, že chceme-li najít původce spamu, stačí nám najít kořen v pomyslném stromě, který má stupeň právě K. Žádný jiný vrchol stejný stupeň už mít nebude, původce je jen jeden.

Ten můžeme najít takto: Postupně projdeme jednotlivé dvojice a budeme si poznamenávat pro každou bytost, kolikrát se vyskytla v nějaké z dvojic. Všimněme si, že tento počet odpovídá právě stupni jejího vrcholu. Potom stačí vyhlásit bytost s právě K výskyty jako původce spamu.



Jak dlouho nám hledání původce potrvá a kolik při tom spotřebujeme paměti? Každou z M dvojic musíme projít právě jednou. Celkově nad tímto procházením strávíme $\mathcal{O}(M)$ času. Potom projdeme každou z N bytostí a zkoumáme počet výskytů. Časová složitost je tedy $\mathcal{O}(N+M)$. Dále si musíme pro každou bytost něco málo pamatovat, paměti tak spotřebujeme $\mathcal{O}(N)$.

Kolik dvojic ale může být celkem? Jelikož každému, až na jediného původce, přišel spam právě jednou, bude těchto dvojic právě $N-1$. Časová složitost nám tedy ve skutečnosti sejde na $\mathcal{O}(N)$.

Václav Končický

29-Z2-6 Devět trpaslíků

Kevin a Zuzka dostanou řadu N čísel v určitém pořadí a u ní mají za úkol rychle odpovídat na určité dotazy. Ty se týkají součtů čísel mezi a-tým a b-tým číslem řady.

Zjevné řešení je pro každý dotaz posloupnost znovu projít a od a-tého do b-tého indexu čísla sčítat. Pokud by se trpaslíci ptali pořád na součet celé řady, procházeli bychom vždy všech N čísel znovu. Kvůli tomu je časová složitost na jeden dotaz $\mathcal{O}(N)$.

Zkusíme zvolit úplně jiný postup. Budeme na něj sice potřebovat více času na přípravu, ale budeme doufat, že se nám to vyplatí. Naším cílem je odpovídat co nejrychleji. Využijeme toho, že sčítání má dobré vlastnosti (například pro násobení by naše řešení nefungovalo), a předpočítáme si čísla, ze kterých budeme schopni rychle vykoukat řešení.

V dalších odstavcích budeme součet čísel mezi a-tým a b-tým indexem značit $s(a, b+1)$. Následujeme tedy běžné programátorské pojetí intervalů, kdy $s(a, b)$ značí součet takového úseku, do kterého a patří ale b už ne.

Představme si, že známe součty na dvou úsecích, které mají společný začátek: $s(a, b)$ a $s(a, c)$. Z těchto čísel jsme schopni odvodit $s(b, c)$ – ten je roven $s(a, c) - s(a, b)$. Rozepište si to. Uvidíte, že se čísla na začátku odečtou a zůstanou jen ta správná.

Tohoto faktu využijeme a spočítáme si součet $s(0, x)$ pro každé x. Úsek začínající na začátku posloupnosti se nazývá prefix posloupnosti, a proto se posloupnosti jejich součtů obvykle říká prefixové součty.

Uvedeme příklad. Máme řadu čísel $1, 5, 4, 3, 6$ a chceme k ní znát všechny prefixové součty. Ty budou vypadat následovně: $0, 1, 6, 10, 13, 19$. Spočítat všechny prefixové součty můžeme průběžně při načítání – každé číslo stačí přičíst k předchozímu, už sečtenému, prefixu, a máme součet o jedna delšího prefixu. Díky tomu nám počítání zabere pouze $\mathcal{O}(N)$.

```
součty[0] = 0
for i in range(N+1):
    součty[i] = součty[i - 1] + čísla[i - 1]
```

Abychom mohli počítat s celou řadou na vstupu, přidáme si na konec součtů ještě jeden prvek navíc, součet celé řady (proto N+1). Všimněte si ale, že čas na přípravu nás vlastně moc nezdržuje – pokud nechceme čísla číst rovnou ze souboru, nevyhneme se načtení do paměti. To už ale trvá stejně dlouho, jako počítání prefixových součtů.

A když už máme pole prefixových součtů připravené, dotazy se zodpovídají velmi snadno. Hodnota součtu čísel mezi a-tým a b-tým je rovna `součty[b + 1] - součty[a]`. Na dotazy tedy umíme odpovídat v konstantním čase, ale potřebujeme $\mathcal{O}(N)$ času na přípravu. Paměťová složitost je $\mathcal{O}(N)$. Pokud se chcete o prefixových součtech a jejich využití dozvědět více, podívejte se na kapitolu kuchařky o intervalových stromech.[3]

Ondra Hlavatý

Výsledková listina druhé série začátečnické kategorie 29. ročníku KSP

	řešitel	*škola*	*ročník*	*sérií*	*Z2-1*	*Z2-2*	*Z2-3*	*Z2-4*	*Z2-5*	*Z2-6*	*série*	*celkem*
0.					8	10	10	12	12	14	66,0	132,0
1.	Jakub Šťastný	G BO-Řeč	2	6	8	10	10	12	12	14	66,0	133,0
2.–3.	Lucia Krajčoviechová	GJHroncaBA	1	2	8	10	10	12	12	14	66,0	132,0
	Petr Šimůnek	G_Hořice	4	2	8	10	10	12	12	14	66,0	132,0
4.–5.	Erik Kučák	GHorMichal	4	2	8	10	10	12	12	14	66,0	131,0
	Daniel Skýpala	GTomkovaOL	−1	6	8	10	10	12	12	13	65,0	131,0
6.	Ondřej Gonzor	G Brandýs	0	5	8	10	10	12	12	14	66,0	130,0
7.	Dávid Šutor	GTerVans	2	2	8	10	10	12	11,5	14	65,5	127,5
8.–9.	Petr Aubrecht	GHeyrovPH	2	2	8	10	10	12	12	8	60,0	126,0
	Jakub Šuráň	GStrážnice	2	2	8	10	10	12	12	14	66,0	126,0
10.	Petr Borňás	G_Roudnice	1	2	8	10	10	12	12	13	65,0	125,0
11.	Michaela Bobeničová	GPošKošice	2	2	8	10	10	12	12		52,0	117,0
12.	Vladimír Chudý	ZŠRonov	0	2	8	10	10	12	12	8	60,0	116,0
13.	Karel Balej	G_Rokycany	2	2	8	10	10	8	12	14	62,0	114,0
14.	Jan Kotovský	GPísnickáPH	−2	2	8	10	8	10	12	8	56,0	112,0
15.	Terézia Strišovská	GJHroncaBA	1	2	8	10	10	12			40,0	106,0
16.	Anna Hollmannová	GSRandyJN	0	6	8	10		6	12	7	43,0	103,0
17.–18.	Michal Kodad	SPŠ_Smíchov	1	2	8	10	10	12	0	14	54,0	101,0
	Kateřina Čížková	G_Rokycany	3	2	8	10	3	12	12	14	59,0	101,0
19.	Andrej Pajtaš	GLitoměřPH	1	2	8	10	10	12	11,5	9	60,5	100,5
20.–21.	Petr Budai	G JGJ PH	0	2	8	10	10	12	12	7	59,0	99,0
	Vincent Orlovský	GTerVans	2	2	8	10	10	4	12	8	52,0	99,0
22.–23.	Vojtěch Březina	GCoubTábor	0	6	8	10			12		30,0	96,0
	Dalibor Kramář	G BO-Řeč	2	2	8	10	10	12	12	8	60,0	96,0
24.	Jakub Ucháč	ŠMaVVzt	1	2	8	10	3	12			33,0	95,0
25.	Vojtěch Hudec	G_ČTřebová	3	5	8	10	10				28,0	94,0
26.	Erik Berta	GAlejKošice	2	2	8	10	1	8	12	6	45,0	89,0
27.	Jan Vodstrčil	G VMýto	0	2	8	10		12	12		42,0	88,0
28.	Jakub Brož	PČGKarVary	3	2	8	10	3	2			23,0	83,0
29.	Andrej Tomči	GHorMichal	4	2	8	10					18,0	82,0
30.	David Nápravník	GLitoměřPH	4	5	8	10	10	6	12		46,0	81,7
31.	Lukáš Caha	GZborovPH	3	2	8	10	10		11		39,0	81,0
32.–33.	Jakub Jirkal	GJungmanLT	2	10	8	10	10	12			40,0	80,0
	Jiří Löffelmann	GLitoměřPH	3	6	8	10	10	12			40,0	80,0
34.	Ondřej Wrzecionko	GTěš	2	2	8	10	3	0			21,0	79,0
35.–36.	Vojtěch Brož	GBudějovPH	2	2	8	10	3	10			31,0	78,0
	Gabriela Pachlová	G_ČTřebová	2	2	8	10	10				28,0	78,0
37.	Petr Macháček	G_TýnNVlt	1	2	8	10					18,0	77,0
38.	Jaroslav Knápek	GLesníZlín	0	2	8	10	0				18,0	74,3
39.	Martin Bencko	GOhradníPH	0	6	8	10	5	4		0	27,0	74,0
40.	Zuzana Urbanová	GFXŠaldyLI	3	2	8	10	3	8	12	4	45,0	73,0
41.	Karel Tomanec	SPŠBruntál	4	2	8	10	10	8			36,0	72,0
42.–43.	Robert Jaworski	GÚstavníPH	−1	5	8	10	1	12			31,0	71,0
	Jozef Mikuláš	CZŠJBosca	0	2	8	10	10		12		40,0	71,0
44.	Lucie Vomelová	GŠpitálsPH	1	2	2	2	3	2	12	8	29,0	69,0
45.	Tomáš Domes	MendelG_OP	4	4	8	10	10	12			40,0	68,0
46.	Ondřej Jamelský	G Cheb	−1	2	8	10	10	12	12		52,0	65,4
47.–48.	Jaroslav Paidar	SPŠMasarLI	3	2	8	10	3	12	12	8	53,0	65,0
	Rajmund Hruška	GPošKošice	4	3							0,0	65,0
49.	Kateřina Nová	G_Vimperk	4	2	8	10	5		12		35,0	63,0
50.–51.	František Kmječ	G Brandýs	1	2	8	10			12	14	44,0	62,0
	Barbora Plačková	GHlu	−1	2	8	10			12		30,0	62,0
52.	Ondřej Krsička	GJarošeBO	1	5							0,0	61,0
53.–55.	Radoslav Hašek	G_Čáslav	3	6	8	10		2			20,0	60,0
	Dominik Pilný	G_Ostrov	2	2	8	10					18,0	60,0
	Martin Sobotka	GLitoměřPH	1	1	8	10	10	12	12	8	60,0	60,0
56.–60.	Daniela Hrbáčová	G Wicht	3	2	4	1	1	2	12		20,0	58,0
	Vojtěch Káně	G Brandýs	1	6	6	10		2			18,0	58,0
	Pavel Martinec	GLesníZlín	3	4	8	10					18,0	58,0
	Filip Masár	PiarGNitra	3	1							0,0	58,0
	Dávid Oravec	G DubNVáh	2	2	8	10					18,0	58,0
61.	Josef Polášek	GKepleraPH	1	2	8	10	3	10			31,0	55,7
62.	Jan Kaifer	GČesBrod	1	8							0,0	55,0
63.	Ondřej Cach	SPSE_Pard	1	5	8	10					18,0	54,0
64.	Michal Mlčoch	G UherBrod	2	3	8	10	10	0			28,0	52,0
65.–66.	Jan Kučera	GFKřižíka	0	2	8	10	3				21,0	49,0
	Ladislav Töpfer	G DrJPekMB	2	2							0,0	49,0
67.–69.	Jan Chybík	SPŠMasarLI	2	2	0	10					10,0	46,0
	Vojtěch Kuchař	ZŠ Sobotka	0	6	8	10					18,0	46,0
	Martin Zmitko	G FrýdlNOs	1	2	8	10					18,0	46,0
70.–71.	Janek Hlavatý	GJirsíkaČB	−2	11	8	10		8			26,0	44,0
	Martin Hofbauer	G BO-Řeč	2	3	8	10					18,0	44,0
72.–73.	Timea Szöllősová	G_Gröss_BA	1	1							0,0	41,0
	Matouš Vondrášek	GJírovcČB	1	1							0,0	41,0
74.–79.	Martin Horáček	GŠumperk	4	1							0,0	40,0
	Martin Melicher	GPošKošice	2	1	8	10	10	12			40,0	40,0
	Adam Perinay	GJHroncaBA	3	1							0,0	40,0
	Dennis Pražák	GJirsíkaČB	2	5							0,0	40,0
	Tomáš Sládek	GJHroncaBA	2	2	8	10					18,0	40,0
	Jakub Szymsza	CmGy PV	1	1							0,0	40,0
80.–81.	Vít Beran	MasG_Plzeň	3	2	4						4,0	38,0
	Jan Štěch	GJirsíkaČB	0	2	4	10		0			14,0	38,0
82.–84.	Jan Juračka	GBystnPern	2	1							0,0	36,0
	Michael Kozel	GZborovPH	3	7	8						8,0	36,0
	Magdaléna Rýdlová	GLesníZlín	3	2	8	10					18,0	36,0
85.	Václav Čermák	GKlatovy	4	1							0,0	34,0
86.–89.	Radim Buráň	G UherBrod	2	2	8	10	10	2			30,0	30,0
	Jan Bíl	GDašickáPA	4	1	8	10	10	2			30,0	30,0
	Tomáš Dulava	GMatOS	3	1	8	10			12		30,0	30,0
	Erik Řehulka	ŠPMNDaGB	1	2	8	10					18,0	30,0
90.	Vladimír Holý	CírkG Plzeň	2	2		1					1,0	29,0
91.–101.	Michal Grňo	BiGyBBHK	4	1							0,0	28,0
	Jaroslav Horáček	GymVyš	3	1							0,0	28,0
	Lucie Kubíčková	GFXŠaldyLI	3	1							0,0	28,0
	Tomáš Nguyen	SPŠÚžlabPH	2	2							0,0	28,0
	Adéla Návratová	ZŠ MTyrš	−1	1							0,0	28,0
	Ondrej Potúček	G_GolNitra	3	3							0,0	28,0
	Vojtěch Poupa	CírkG Plzeň	−1	1							0,0	28,0
	Adrián Rošinec	GHorMichal	4	1							0,0	28,0
	Eliška Vlčinská	GHladnov	2	2							0,0	28,0
	Tomáš Vítek	G_Břeclav	0	1							0,0	28,0
	Benedikt Žour	G UherBrod	2	8		4					4,0	28,0
102.	Jiří Janoušek	GBudějovPH	1	2	2						2,0	27,0
103.	Jan Martínek	GTomkovaOL	2	1							0,0	26,0

[3] http://ksp.mff.cuni.cz/viz/kucharky/intervalove-stromy