

Komentáře k páté sérii třicátého sedmého ročníku KSP

37-5-3 Ledová socha

Vzorové řešení této úlohy běží v čase $\mathcal{O}(N^{2.5})$ pro mapu velikosti $N \times N$. Zde poslyšte jedno ještě efektivnější, inspirované myšlenkami Alberta Bakoče, Erika Ježka a Ríši Hladíka.

Nejprve si všimneme, že každou cestu sochy můžeme rozdělit na svislé a vodorovné úseky, které jsou na sebe napojeny políčky, v nichž má socha nulovou rychlost (to splňuje i počáteční a koncové políčko cesty). Vytvoříme tedy graf, jehož vrcholy budou políčka mapy a z políčka x do y povede hrana délky t právě tehdy, dá-li se z x s nulovou rychlostí dostat svisle nebo vodorovně do y s nulovou rychlostí za t kroků.

Tento graf má N^2 vrcholů a $\mathcal{O}(N^3)$ hran (vrcholy mají stupeň až N). Chceme v něm hledat nejkratší cestu, která určitě nebude delší než $2N^2$ kroků, neboť na sousední políčko se umíme dostat za 2 kroky.

Nabízí se použít Dijkstrův algoritmus, ale ten by měl aspoň kubickou složitost (kvůli počtu hran). Místo toho budeme indukci podle času počítat, která políčka jsou v daném čase dosažitelná z počátku. To vlastně dělá i Dijkstrův algoritmus, ale my se obejdeme bez zkoumání všech hran.

Jak vypadají hrany

Pojďme nejprve prozkoumat, jak vypadají hrany, které vedou z nějakého políčka. Ze čtyř možných směrů si bez újmy na obecnosti vybereme směr doprava, ostatní směry vyřešíme analogicky. Pohybujeme se tedy v nějakém vodorovném úseku bludiště až k nejbližší překážce vpravo.

Rozmysleme si, jak vypadá cesta o nejméně krocích, která využije maximální rychlost v . Tato cesta postupně nabývá rychlostí $1, 2, \dots, v-1, v, v-1, \dots, 1, 0$. Trvá tedy $2v$ kroků a dostane do vzdálenosti $2 \cdot (1+2+\dots+(v-1)) + v = 2 \cdot v(v-1)/2 + v = v^2$.

Další cesty s maximální rychlostí v vzniknou vkládáním kroků, kdy udržujeme stejnou rychlost. Vložíme-li jeden krok, dostaneme pro jeho různé polohy vzdálenosti $v^2+1, v^2+2, \dots, v^2+v$, pokaždé na $2v+1$ kroků. Vložíme-li dva kroky, můžeme dostat libovolnou vzdálenost od v^2+2 do v^2+2v na $2v+2$ kroků. Vkládat více než dva kroky se už nikdy nevyplatí, protože vzdálenost $v^2+2v+1 = (v+1)^2$ už můžeme získat cestou s maximální rychlostí $v+1$ za $2v+2$ kroků.

Obrátíme-li tuto úvahu, dostaneme, že za $2v$ kroků se můžeme dostat do vzdálenosti v^2 cestou o maximální rychlosti v , a do libovolné menší vzdálenosti cestou s menší rychlostí, do které jsme doplnili čekací kroky. A za $2v+1$ kroků se umíme dostat do vzdálenosti v^2+v .

Intervaly a jejich srážky

Oblast, kam se z daného políčka (budeme mu říkat *semínko*) dostaneme za k kroků vpravo, je tedy nějaký interval, který začíná v semínku a s rostoucím k roste doprava čím dál větší rychlostí. Časem se možná zastaví o nějakou překážku.

Co kdybychom časem do téhož řádku zasadili další semínko? Z něj by se začal rozšiřovat doprava další interval a po nějaké době by intervaly mohly narazit do sebe.

Pokud levý interval pochází z novějšího semínka než pravý, poroste levý pomaleji než pravý, takže bude levý navěky schovaný pod pravý, a tudíž na levý můžeme zapomenout. Totéž nastane, pokud jsou intervaly stejně staré.

Pokud je naopak levý interval novější, bude sice nějakou dobu schovaný pod pravým, ale jelikož levý roste rychleji, časem prorazí skrz pravý interval, a tehdy budeme moci pravý interval zapomenout.

Teď se podívejme na obecnou situaci. V nějakém řádku (respektive jeho části mezi překážkami) máme několik semínek, z každého roste doprava interval. Intervaly jsou buď viditelné, nebo dočasně zakryté, nebo natrvalo zakryté. Ty natrvalo zakryté si nemusíme pamatovat.

Poskočí-li čas o jeden krok, každý interval se o kousek zvětší. Viditelný interval objeví nová políčka, dočasně zakrytý interval se může probourat následujícím intervalem.

Zvětšování viditelných intervalů můžeme naučtovat nově objeveným políčkům, případně narazí-li interval do překážky, naučtujeme ho překážce a pak už ho smažeme.

S dočasně zakrytými je to horší: při každém kroku v čase jich můžeme projít hodně, aniž by se cokoliv zajímavého stalo. Proto si pro každý takový interval nastavíme budík na čas, kdy tento interval prorazí ten vpravo od něj; tento čas dovedeme spočítat konstantně rychle. Budíky si můžeme udržovat v poli indexovaném časem (ten je nejvýš $2N^2$), v každém okénku pole bude spojový seznam budíků.

Průběh celého algoritmu

Celý algoritmus začne tím, že si rozdělíme políčka bludiště do souvislých úseků řádků a sloupečků. Pro každý úsek si pořídíme seznam intervalů, které bude udržovat již popsáním způsobem. Abychom uměli rychle najít intervaly, které jsou zrovna viditelné, pořídíme si navíc jejich globální seznam. Pole budíků nám postačí jedno společné.

V čase 0 jsme na počátečním políčku a zasadíme semínko intervalů pro každý ze čtyř směrů.

Pokaždé, když se čas posune o 1 krok, rozšíříme všechny intervaly, což zahrnuje i zazvonění všech budíků přiřazených k aktuálnímu času.

Rozšíření intervalu v nějakém směru může objevit nové políčko. V takovém případě v něm zasadíme semínka v ostatních směrech. To je trochu zrádné, protože potřebujeme založit nový interval a správně ho zatřídit do seznamu stávajících intervalů. Proto v seznamech intervalů budeme udržovat i všechna dosud neobjevená políčka jako intervaly velikosti 1, které zatím nerostou. Každé políčko si bude pro každý směr pamatovat odkaz na příslušný interval.

Čas zastavíme v okamžiku, kdy objevíme cílové políčko, případně když dojdou intervaly, které by se ještě mohly rozšiřovat.

Analýza složitosti

Celkem zasadíme $\mathcal{O}(N^2)$ semínek, každé v konstantním čase. Růst intervalů už umíme účtovat nově objeveným políčkům, takže celkem trvá $\mathcal{O}(N^2)$. Přitom vznikne $\mathcal{O}(N^2)$ budíků, každý z nich ošetříme v konstantním čase.

Jeden krok v čase obnáší zpracování zazvonivších budíků a rozšíření všech viditelných intervalů. Práci na budíky a

intervaly jsme už započítali, zbývá $\mathcal{O}(1)$ práce na krok, celkem tedy $\mathcal{O}(N^2)$.

Celkově jsme naučtovali $\mathcal{O}(1)$ času každému políčku, $\mathcal{O}(1)$ každému budíku a $\mathcal{O}(1)$ každému kroku, což dohromady činí $\mathcal{O}(N^2)$.

Martin „Medvěd“ Mareš