

Angelas

Pirmiausia pastebėkime, kad angelo MP skaičius nėra esminis. Mums svarbu, kiek angelas gali atlikti šuolių apskritai.

Kadangi $MP/\delta MP \leq 10$, tai šuolių skaičius nėra didesnis nei 10.

Kiekvieną angelo būseną galime apibūdinti keturiais dydžiais: eilute ir stulpelių matricoje, atliktų šuolių skaičiumi ir HP . Susidarome grafą, kurio viršūnė apibūdinama eilute ir stulpelių matricoje ir atliktų šuolių skaičiumi.

Galime pritaikyti Dijkstros algoritmą. Kiekvieną kartą išrenkame neaplankytą viršūnę su maksimaliu HP .

Jei tą viršūnę (v) būtume galėję pasiekti su didesniu HP , turėtų egzistuoti kita viršūnė (u), iš kurios būtume atėję.

Jei u jau paimta, tai v viršūnės HP jau būtų atnaujintas ir didesnis - prieštara.

Jei u dar nepaimta, tai jos turimas HP nėra didesnis už viršūnės v HP , turint omenyje tai, kad HP keliaujant gali tik mažėti, tai eidami iš viršūnės u į viršūnę v negausime didesnio HP - prieštara.

Vadinasi v viršūnė pasiekama su maksimaliu HP . Ją pažymime kaip aplankytą ir atnaujiname visų kitų viršūnių, pasiekiamų iš v , gyvybės taškus HP .

Perėjimai iš viršūnių išplaukia tiesiogiai iš angelo galimybių. Jei kalva kitoje viršūnėje nėra didesnė, angelas gali į ją pereiti (nenaudodamas skrydžių). Kitu atveju, jis gali sunaudoti vieną savo skrydį.

Kiekvienai viršūnei radome maksimalius galimus HP .

Žinome eilutę ir stulpelį, kuriame turi atsistoti angelas. Perrenkame visus įmanomus šuolių skaičius. Pasirenkame mažiausią šuolių skaičių, kuris užtikrina maksimalų HP skaičių iš visų galimų pasirinkimų.

Jei $HP < 0$, angelas keliauti negali.

Viršūnių skaičius yra $100 \cdot 100 \cdot 10 = 100\,000$.

Jei Dijkstros algoritme naudojama prioritetinga eilė, gaunamas $O(n \log n)$ sprendimas. Kitu atveju, gaunamas $O(n^2)$ sprendimas.