1 Popis řešení

K nalezení cesty použijeme modifikovaný Dijkstrův algoritmus. Pravděpodobnosti přepadení budeme chápat jako ceny hran a hledat budeme nejkratší cestu, ovšem při aktualizaci vrcholů nebudeme pro výpočet vzdálenosti od počátku hrany přičítat, ale počítat výslednou pravděpodobnost pomocí principu inkluze a exkluze. Pravděpodobnost přepadení na nejbezpečnější cestě získáme, když zpracujeme cíl. Pokud chceme najít nejen nejmenší pravděpodobnost, ale zároveň i cestu s touto pravděpodobností, algoritmus dále modifikujeme například tak, že každý vrchol si při přidání na prioritní frontu bude pamatovat i vrchol, který ho tam přidal.

2 Pseudokód

```
foreach v je vrchol
  d[v] = nekonečno
  pred[v] = null
d[start] = 0

while halda není prázdná {
  v = ExtractMin()
  if v == cil break
  foreach w soused od v
    alt = d[v] + vw - d[v]*vw
    if (d[w] > alt)
      d[w] = alt
      pred[w] = v
}

cesta = ""
v = cil
while v != start
  cesta = pred[v] + cesta
  v = pred[v]
```

3 Důkaz správnosti

Nejprve je potřeba zjistit, zda Dijkstrův algoritmus můžeme na takovýto graf použít. První podmínkou pro tento algoritmus je nezápornost všech ohodnocení. Tuto podmínku máme splněnou triviálně z definice pravděpodobnosti. Také si můžeme uvědomit, že žádný sled, který není cestou, nebude bezpečnější než cesta, tedy nikdy si nepomůžeme cyklem. Toto je taktéž zřejmé; pokud jsou všechny hrany cyklu ohodnoceny nulou, výsledná pravděpodobnost se nijak nezmění. Při jakýchkoliv nenulových hodnotách se pak nutně zvýší. Odtud a z faktu, že algoritmus vybírá vždy nejbližší vrchol, můžeme vyvodit, že při extrakci cíle může algoritmus skončit.

Je také nutno poukázat na to, že pokud známe pravděpodobnost přepadení do předposledního vrcholu nějaké cesty a ohodnocení poslední hrany, nemusíme počítat výslednou pravděpodobnost celou znovu pomocí principu inkluze a exkluze, ovšem dá se vyjádřit jednoduše pomocí výrazu $a + b - ab$.

Toto tvrzení dokážeme následovně. Z principu inkluze a exkluze víme, že pravděpodobnost přepadení na cestě o dvou hranách s pravděpodobnostmi a a b je rovna $a + b - ab$. Jelikož známe výslednou pravděpodobnost přepadení na prvních $n - 1$ hranách, můžeme pro účely našeho výpočtu nahradit tyto hrany jedinou hranou ohodnocenou touto výslednou pravděpodobností. Poté už máme cestu složenou ze dvou hran a můžeme využít vzorec uvedený výše.

Jiný náhled na ospravedlnění použití Dijkstrova algoritmu je ten, že zatímco klasická varianta hledá minima z délek cest, tato verze hledá minima z celkových pravděpodobností cest (získaných pomocí PIE).

4 Časová složitost

Použili jsme Dijkstrův algoritmus modifikovaný pouze o konstantní množství operací, tedy časová složitost algoritmu bude v závislosti na implementaci $O(n^2 + m)$ (pro pole), $O((n + m) \log n)$ (pro binární haldu) nebo $O(m + n \log n)$ (pro Fibonacciho haldu).

5 Prostorová složitost

Pamatujeme si konstantní množství informací o každém vrcholu i o každé hraně. Celková prostorová složitost je tedy $O(n + m)$.