Pravdepodobnostné algoritmy

Algoritmy, ktoré využívajú **náhodné čísla.**

Las Vegas algoritmy.

- Vždy dajú správnu odpoveď.
- Náhodné čísla ovplyvňujú čas $\Rightarrow$ **očakávaná časová zložitosť**

Monte Carlo algoritmy.

- Bežia vždy rýchlo.
- Občas dajú nesprávnu odpoveď $\Rightarrow$ **pravdepodobnosť chyby** p
  - Jednostranné chyby
    (napr. "áno" je vždy dobre, "nie" môže byť chybné)
  - Obojstranné chyby

Dôležité: Rýchlosť/chybovosť algoritmu **nezávisí od vstupu**, ale len od výberu náhodných čísel! (t.j. neexistuje "zlý" vstup)

Kruskalov algoritmus (1956)

```
MST-KRUSKAL(E):
  repeat:
    (u,v) := hrana s minimálnou cenou
    T := T + {(u,v)}
    skontrahuj hranu (u,v)
```

Časová zložitosť: $O(m \log n)$
(použi dátovú štruktúru pre UNION/FIND-SET)

Primov algoritmus (1957)

```
MST-PRIM(E):
  s := ľubovoľný počiatočný vrchol
  repeat
    (s,v) := hrana s minimálnou cenou z vrcholu s
    T := T + {(s,v)}
    skontrahuj hranu (s,v)
```

Časová zložitosť: $O(m \log n)$

ak použijeme Fibonacciho heap: $O(n \log n + m)$

Borůvkov algoritmus (1926)

```
MST-BORUVKA(E):
  repeat
    pre každý vrchol v[i] nájdi z neho vychádzajúcu hranu
    e[i] s minimálnou cenou;
    T := T + {e[1],e[2],...}
    skontrahuj hrany e[1],e[2],...
```

Časová zložitosť: $O(m \log n)$

(v každom kroku minimálne polovicu vrcholov odstránime)

Pravdepodobnostný algoritmus (Karger et al. 1994)

```
MST-RANDOMIZED(E):
  repeat
1:   urob 2x Borůvkov krok
2:   R := náhodný podgraf (každá hrana s pravdepodobnosťou p)
3:   F := MST-RANDOMIZED(R)
4:   H := ťažké hrany z E vzhľadom ku F
5:   E := E - H
```

```
F:={}; R:={}
pre všetky hrany e od najmenšej po najväčšiu:
  ak e neurobí cyklus v F, OZNAČKUJ hranu
  s pravdepodobnosťou p:
     R:=R U {e}
     ak je e označkovaná, F:=F U {e}
```

R je náhodný podgraf z kroku 2

F je jeho najlacnejšia kostra

Len označkované hrany môžu byť ľahké