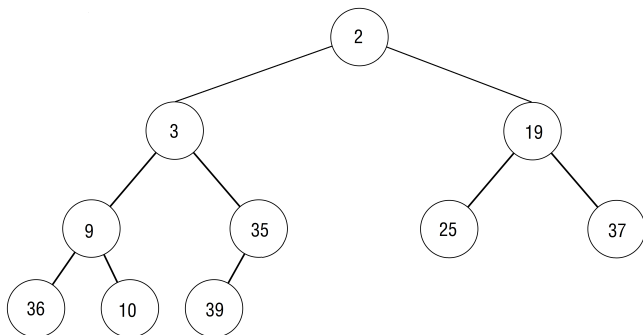


EADŠ - cvičenie 3

5. októbra 2023

Halda - opakovanie

- ▶ implementácia dátovej štruktúry *priority queue*
- ▶ poloutriedený úplný binárny strom
- ▶ minimum v koreni
- ▶ rodič má menší kľúč (menšiu prioritu) ako potomok



Priority queue - opakovanie

Abstrakcia nad haldou. Má operácie:

1. `insert(x)`
2. `extractMin()`

Vieme ju používať v rôznych algoritmoch, a pomocou nej vytvárať iné dátové štruktúry.

Pravdepodobnosti:

1. Nieкто hádza mincou. $P(1) = \frac{1}{2}$, $P(0) = \frac{1}{2}$, ja tipujem náhodne. Aký je očakávaný počet hodov v ktorých sa trafím?

Pravdepodobnosti:

1. Niekoľko hádže mincou. $P(1) = \frac{1}{2}$, $P(0) = \frac{1}{2}$, ja tipujem náhodne. Aký je očakávaný počet hodov v ktorých sa trafím?
2. Niekoľko hádže mincou. $P(1) = \frac{1}{3}$, $P(0) = \frac{2}{3}$, ja tipujem náhodne. Aký je očakávaný počet hodov v ktorých sa trafím?

Pravdepodobnosti:

1. Niekoľko hádže mincou. $P(1) = \frac{1}{2}$, $P(0) = \frac{1}{2}$, ja tipujem náhodne. Aký je očakávaný počet hodov v ktorých sa trafím?
2. Niekoľko hádže mincou. $P(1) = \frac{1}{3}$, $P(0) = \frac{2}{3}$, ja tipujem náhodne. Aký je očakávaný počet hodov v ktorých sa trafím?
3. Niekoľko hádže mincou, strieda 0 a 1. Aký je očakávaný počet hodov v ktorých sa trafím?

Pravdepodobnosti:

1. Niekoľko hádže mincou. $P(1) = \frac{1}{2}$, $P(0) = \frac{1}{2}$, ja tipujem náhodne. Aký je očakávaný počet hodov v ktorých sa trafím?
2. Niekoľko hádže mincou. $P(1) = \frac{1}{3}$, $P(0) = \frac{2}{3}$, ja tipujem náhodne. Aký je očakávaný počet hodov v ktorých sa trafím?
3. Niekoľko hádže mincou, strieda 0 a 1. Aký je očakávaný počet hodov v ktorých sa trafím?
4. Ako chcem tipovať, ak viem, že sa jedná o prípad 2. ?

Triedenie (sort)

- ▶ zorad'ujeme prvky (od najmenšieho po najväčšie)
- ▶ ako o jednom z mála problémov v informatike vieme dokázať, dolný odhad zložitosti
- ▶ Algoritmy:
 - ▶ pomalé - $O(n^2)$ (min sort, insertion sort, ...)
 - ▶ rýchle - $O(n \log n)$ (quicksort, merge sort, heapsort, ...)
 - ▶ špeciálne - $O(n)$ (counting sort radix sort, bucket sort, ...)

Triedenie - 1

Máme pole s celými číslami ($[5, 1, 3, 2, 1, 2, 2]$).

Chceme pre každé číslo napísať, koľko krát sa v poli vyskytuje ($\{1:2, 2:3, 3:1, 5:1\}$).

Triedenie - 2

Anagram alebo prešmyčka, je slovo vytvorené preskupením písmen iného slova (napr. lampá → palma).

Máme zoznam existujúcich slov (slovenský slovník), a chceme:

1. pre zadané slovo zistiť počet anagramov
2. nájsť také slovo, ktoré má najviac anagramov

Triedenie - 3

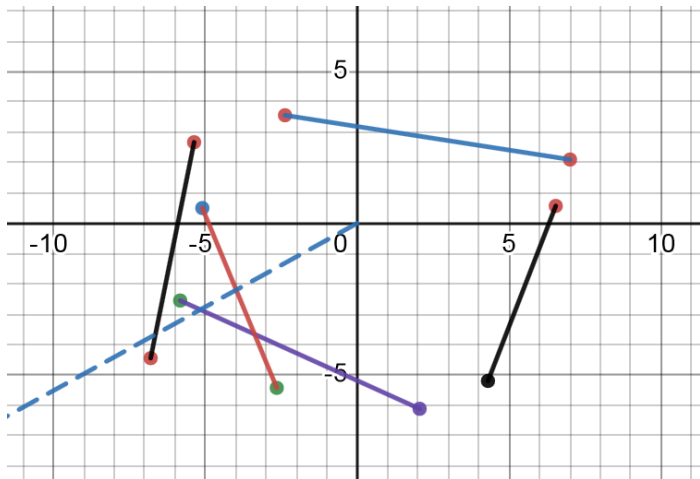
Robíme rozvrh pre školu. Máme n prednášok, i -ta prednáška má čas začiatku a konca (s_i, f_i) .

Chceme zistiť, koľko najmenej miestností potrebujeme, aby sa všetky prednášky vedeli konať, a žiadne dve neboli naraz v tej istej miestnosti.

Triedenie - 4

Lukostrelec strieľa z bodu $[0, 0]$ ľubovoľným smerom. Koľko najviac vie zasiahnuť terčov jedným šípom? (Šíp po náraze do terča pokračuje ďalej rovnakým smerom.)

Triedenie - 4



sort vs. sorted

```
a = list(range(50))  
a.sort()
```

```
b = list(range(50))  
sorted(b)
```

sort vs. sorted

```
a = list(range(50))  
a.sort()
```

```
b = list(range(50))  
b = sorted(b)
```

Triedenie zostupne

```
a = list(range(50))  
a.sort(reverse=True)
```

```
b = list(range(50))  
b = sorted(a, reverse=True)
```


Triedenie podľa vlastného kľúča

```
a = ['abc', 'ABCD', 'bcd']  
a.sort()  
# ['ABCD', 'abc', 'bcd']
```

```
a.sort(key=str.lower)  
# ['abc', 'ABCD', 'bcd']
```

```
sorted(a, key=str.lower)  
# ['abc', 'ABCD', 'bcd']
```

Triedenie podľa vlastnej porovnávacej funkcie

```
from functools import cmp_to_key

def compare(item1, item2):
    return (item1-15)**2 - (item2-15)**2

a = list(range(50))

a.sort(key=cmp_to_key(compare))

sorted(a, key=cmp_to_key(compare))
```

Vlastná porovnávací funkcia

- ▶ záporná hodnota (< 0) keď `item1` má byť vo výslednom poli **pred** `item2`
- ▶ kladná hodnota (> 0) keď `item1` má byť vo výslednom poli **za** `item2`
- ▶ nulová hodnota (0) keď `item1` má byť vo výslednom poli **rovnako** ako `item2` (sú rovné)