

Greedy algoritmus pre problém výberu aktivít

Sort all activities by their finishing time
(now $f[1] \leq f[2] \leq \dots \leq f[n]$)

```
last_activity_end := -infinity;
```

```
for i:=1 to n
```

```
    if (s[i] >= last_activity_end) then
```

```
        output activity (s[i], f[i]);
```

```
        last_activity_end := s[i];
```

Časová zložitosť: $\Theta(n \log n)$

Typický greedy algoritmus

- Každé riešenie získame pomocou postupnosti rozhodnutí.
- Nie všetky rozhodnutie vedú k optimálnemu riešeniu.
- V každom kroku:
 - Ováhuju všetky možné rozhodnutia pomocou nejakej váhovacej funkcie.
 - Vyber rozhodnutie s najväčšou váhou.

“Vzor” dôkazu správnosti greedy algoritmu

Lema: Predpokladajme, že greedy algoritmus vráti riešenie G . Potom existuje optimálne riešenie, ktoré sa s riešením G zhoduje na prvých k voľbách.

Dôkaz: Matematickou indukciou podľa k .

Báza indukcie. Pre $k = 0$ – ľubovoľné optimálne riešenie.

Indukčný krok. (Predpokladajme, že sme neurobili chybu pri prvých k voľbách, potom aj $(k + 1)$ -vá voľba je OK.)

- Predpokladajme, že existuje optimálne riešenie OPT , ktoré sa zhoduje s G na prvých k voľbách.
- Vyrobíme riešenie OPT' :
 - OPT' má rovnakú hodnotu ako OPT
(a preto je tiež optimálne)
 - OPT' súhlasí s G na jednej ďalšej $(k + 1)$ -vej voľbe.

Greedy algoritmus pre Huffmanov strom

Compute frequencies of all characters in S

F:=empty-forest;

for all characters x in the alphabet do

 T:=new leaf(x);

 add T to F;

while F contains more than one tree do

 T1:=extract tree with minimum frequency from F;

 T2:=extract tree with minimum frequency from F;

 T:=new tree where T1 is a left child
 and T2 is a right child;

 add T to F;

return F;