

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;

“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.

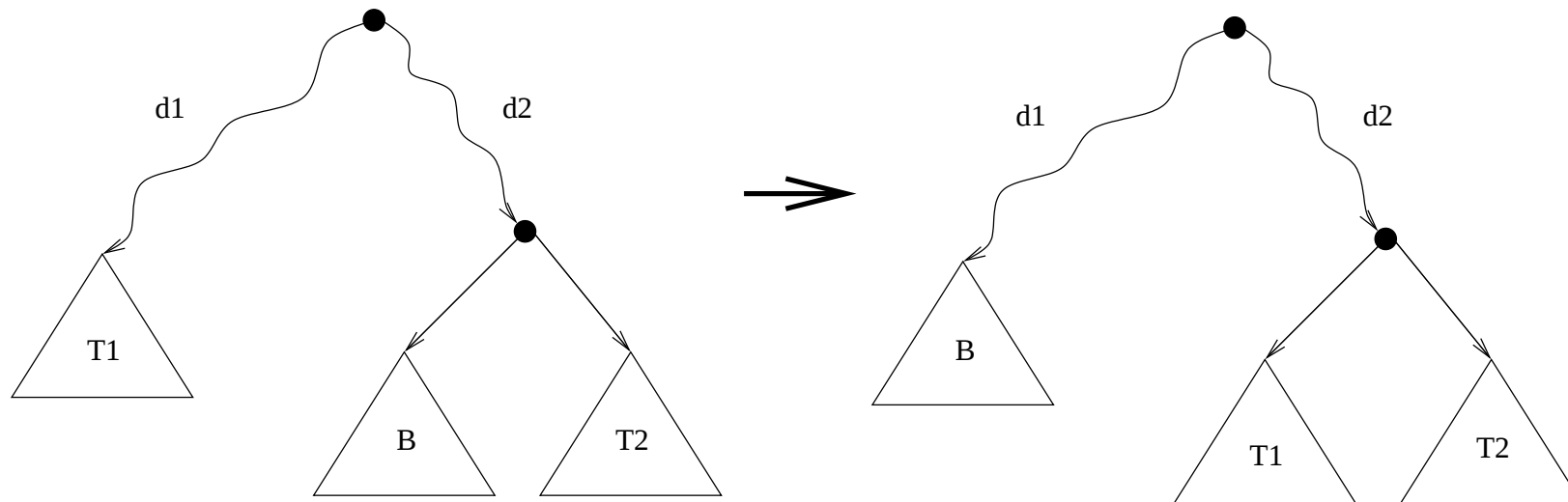
Dôkaz: Matematickou indukciou podľa k .

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.
- Vyročíme riešenie OPT' :
 - OPT' má rovnakú hodnotu ako OPT
(a preto je tiež optimálne)
 - OPT' súhlasí s G na jednej ďalšej $(k + 1)$ -vej voľbe.

Lema: Nech greedy algoritmus po i krokoch vytvoril les $F = (T_1, T_2, \dots, T_k)$. Potom existuje optimálny kódovací strom, ktorý obsahuje $T_1, \dots, T_k$ ako podstromy.



Rozmieňanie peňazí

```
coins[0]:=0;
for i:=1 to S do
  min:=infinity;
  for j:=1 to m do
    if d[j]<=i and coins[i-d[j]]<min then
      min:=coins[i-d[j]];
  coins[i]:=1+min;

return coins[S];
```

Time: $\Theta(mS)$

Rozmieňanie peňazí s vypísaním riešenia

```
coins[0]:=0;
for i:=1 to S do
  min:=infinity;
  for j:=1 to m do
    if d[j]<=i and coins[i-d[j]]<min then
      min:=coins[i-d[j]];
  *   minchoice:=j;
  coins[i]:=1+min;
  * choice[i]:=minchoice;

if coins[S]=infinity then write 'No solution!';
else
  while S>0 do
    write d[choice[S]];
    S:=S-d[choice[S]];
```

Dynamické programovanie—zhrnutie

1. **Urcíme podproblém.**

- aké sú rozmery matice, ktorú budeme vyplňať?
- aký je presný význam každého políčka matice?
- kde v matici nájdeme riešenie pôvodnej úlohy?

2. **Vyriešime podproblém za pomoci iných podproblémov.**

Ako vypočítame jedno políčko matice z iných políčiek matice?

3. **Bázové podproblémy.** Ktoré políčka nemožno vypočítať pomocou vzťahov z predchádzajúceho kroku? Aké hodnoty by mali obsahovať?

4. **Vyberieme poradie vyplňania.** V akom poradí musíme maticu vyplňať tak, aby sme v každom kroku mali vypočítané všetky políčka, ktoré potrebujeme na výpočet daného políčka?