Dynamické programovanie, rozdeľuj a panuj

21. októbra 2014

Dynamické programovanie—zhrnutie

1. *Určíme podproblém.*
    - *aké sú rozmery matice, ktorú budeme vypĺňať?*
    - *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 vypĺňania. V akom poradí musíme maticu vypĺňať tak, aby sme v každom kroku mali vypočítané všetky políčka, ktoré potrebujeme na výpočet daného políčka?

Najkratšia triangulácia

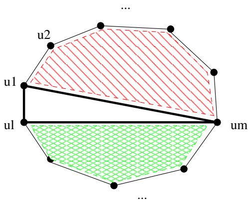
Problém

Daný je konvexný n-uholník (vymenovaním vrcholov v smere hodinových ručičiek $v_1, v_2, \ldots, v_n$). Nájdite trianguláciu s najmenšou dĺžkou.

Najktratšia triangulácia

Podproblém

$t[u_1, \ldots, u_l]$ –najkratšia triangulácia n-uholníka $(u_1, u_2, \ldots, u_l)$, kde $u_1, \ldots, u_l$ je podpostupnosť $v_1, \ldots, v_n$.

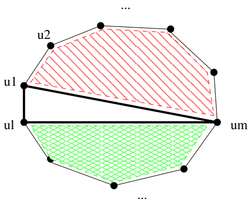


$$d(u_1, u_l) + t[u_1, \ldots, u_m] + t[u_m, \ldots, u_l]$$

Najktratšia triangulácia

Podproblém

$t[u_1, \ldots, u_l]$ –najkratšia triangulácia n-uholníka $(u_1, u_2, \ldots, u_l)$, kde $u_1, \ldots, u_l$ je podpostupnosť $v_1, \ldots, v_n$.



$$d(u_1, u_l) + t[u_1, \ldots, u_m] + t[u_m, \ldots, u_l]$$

$$t[u_1, \ldots, u_l] = \min_{1<m<l} \{d(u_1, u_l) + t[u_1, \ldots, u_m] + t[u_m, \ldots, u_l]\} \quad (1)$$

Najkratšia triangulácia — algoritmus

```
// base case - j=i+1
for i:=1 to n-1 do
  T[i,i+1]:=D[i,i+1];

for delta:=2 to n-1 do
  // cases where j-i=delta
  for i:=1 to n-delta do
    j:=i+delta; T[i,j]:=infinity;
    // try all possible triangles v_i,v_j,v_m
    for m:=i+1 to j-1 do
      cost:=D[i,j]+T[i,m]+T[m,j];
      if cost<T[i,j] then
*       T[i,j]:=cost;


return T[1,n];
```

Najkratšia triangulácia — algoritmus

```
// base case - j=i+1
for i:=1 to n-1 do
  T[i,i+1]:=D[i,i+1];

for delta:=2 to n-1 do
  // cases where j-i=delta
  for i:=1 to n-delta do
    j:=i+delta; T[i,j]:=infinity;
    // try all possible triangles v_i,v_j,v_m
    for m:=i+1 to j-1 do
      cost:=D[i,j]+T[i,m]+T[m,j];
      if cost<T[i,j] then
        T[i,j]:=cost;
        M[i,j]:=m;

return T[1,n];
```

Najkratšia triangulácia — vypísanie riešenia

```
function give_solution(i,j)
  output edge (i,j);
  if j>i+1 then
    give_solution(i,M[i,j]);
    give_solution(M[i,j],j);
```

Rozdeľuj a panuj

Merge sort—hlavný program

```
// sort sequence A[l..r]
function merge_sort(l,r)
  // base case - 1 element is always sorted
  if (l=r) then return;
  m=(l+r) div 2;
  // we need to sort sequences l..m, m+1..r
  merge_sort(l,m);
  merge_sort(m+1,r);
  // and finally merge two sorted sequences
  merge(l,m,r);
```

Merge sort—merge

```
//merge two sorted sequences l..m, m+1..r
function merge(l,m,r)
  copy A[l..m] to L; L[m-l+2]:=infinity;
  copy A[m+1..r] to R; R[r-m+1]:=infinity;

  i:=1; j:=1; k:=l;
  while (L[i]<infinity or R[i]<infinity) do
    if L[i]<=R[j] then
      A[k]:=L[i];
      i:=i+1; k:=k+1;
    else
      A[k]:=R[j];
      j:=j+1; k:=k+1;
```

Rozdeľuj a panuj

Rozdeľuj. Rozdeľ problém na niekoľko menších podproblémov.

Panuj. Každý podproblém vyrieš samostatne rekurzívnym volaním.
Ak sú podproblémy dostatočne malé, vyrieš ich priamočiaro.

Kombinuj. Skombinuj riešenia menších podproblémov do riešenia pôvodného veľkého problému.

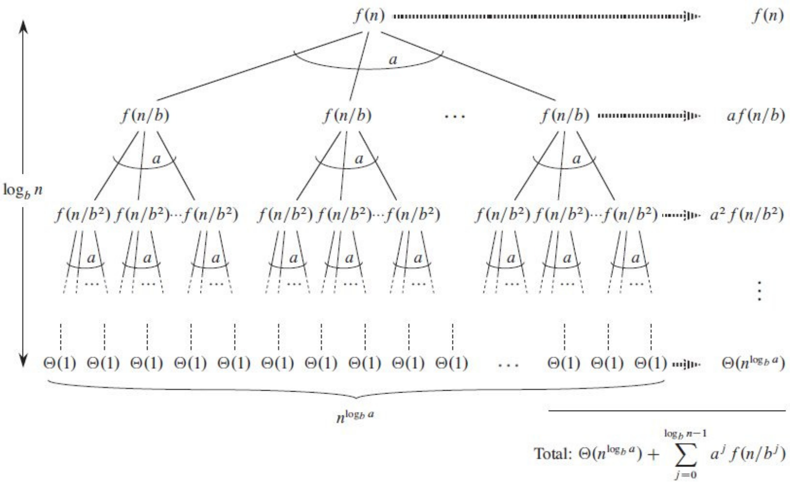
Theorem

Hlavná veta (master theorem): Nech $T(n) = aT(n/b) + f(n)$, $T(1) = \Theta(1)$. *Nech* $k = \log_b a$. *Potom:*

1. *Ak* $\boxed{f(n) \in O(n^{k-\varepsilon})}$ *pre niektoré* $\varepsilon > 0$, *potom* $\boxed{T(n) \in \Theta(n^k)}$.
2. *Ak* $\boxed{f(n) \in \Theta(n^k)}$, *potom* $\boxed{T(n) \in \Theta(f(n) \log n)}$.
3. *Ak* $\boxed{f(n) \in \Omega(n^{k+\varepsilon})}$ *pre niektoré* $\varepsilon > 0$ *a platí podmienka regularity, potom* $\boxed{T(n) \in \Theta(f(n))}$.

Podmienka regularity: Existuje $c < 1$ *také, že pre všetky dostatočne veľké* n *platí* $af(n/b) \leq cf(n)$.

Poznámka: Veta platí aj v prípade rozumných usporiadaní dolných a horných celých častí - viď napr. CLRS2 4.4.2.


$f(n)$
a
$f(n/b)$ $f(n/b)$ ... $f(n/b)$
$af(n/b)$
$\log_b n$
$f(n/b^2)$
$a^2 f(n/b^2)$
$\Theta(1)$
$\Theta(n^{\log_b a})$
$n^{\log_b a}$
Total: $\Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)$