

طراحی الگوریتم ها

روش شاخه و حد

استاد درس: مهدی جبل عاملی

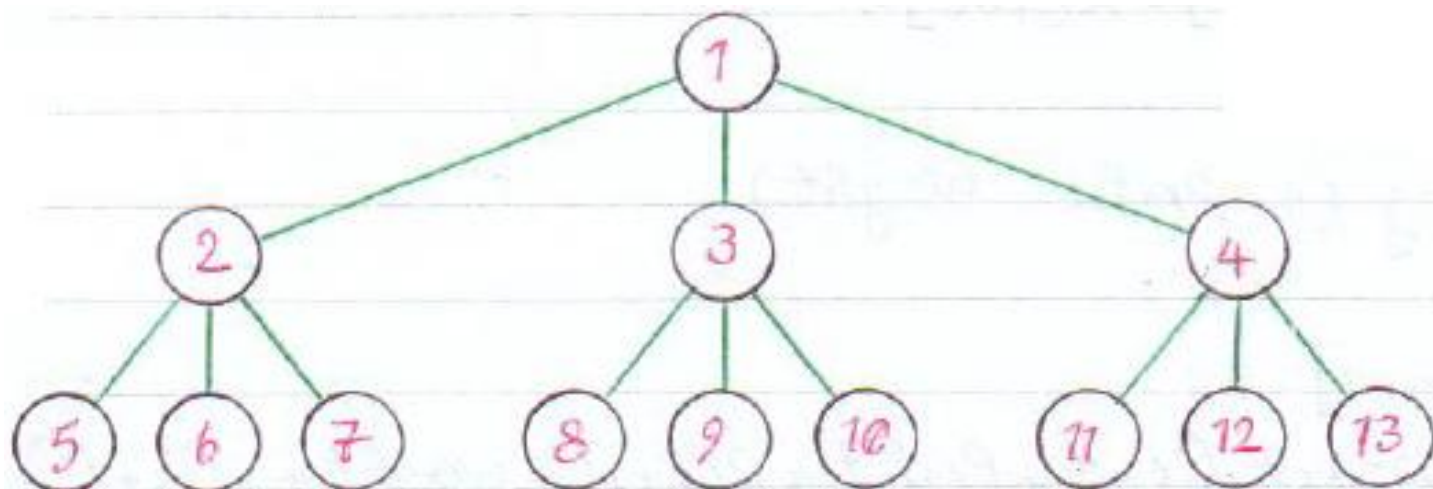
شاخه و حد (Branch & Bound)

شاخه و حد (Branch & Bound)

انواع پیمایش درخت

- پیمایش عمقی (Depth-first)
- پیمایش سطحی (Breadth-first)
-

شاخه و حد (Branch & Bound)



انواع پیمایش درخت

- پیمایش عمقی (Depth-first)
- پیمایش سطحی (Breadth-first)
-

Branch & Bound - Breadth-First (فرست گستره) {

Initialize (Q);

V = root of Tree

Enqueue (Q, V);

while not Empty (Q)

{

V = Dequeue (Q);

for each child U of V

{

if value(U) is better than Best

Best = value(U);

if Bound(U) is better than Best

Enqueue (Q, U);

}

}
return Best;

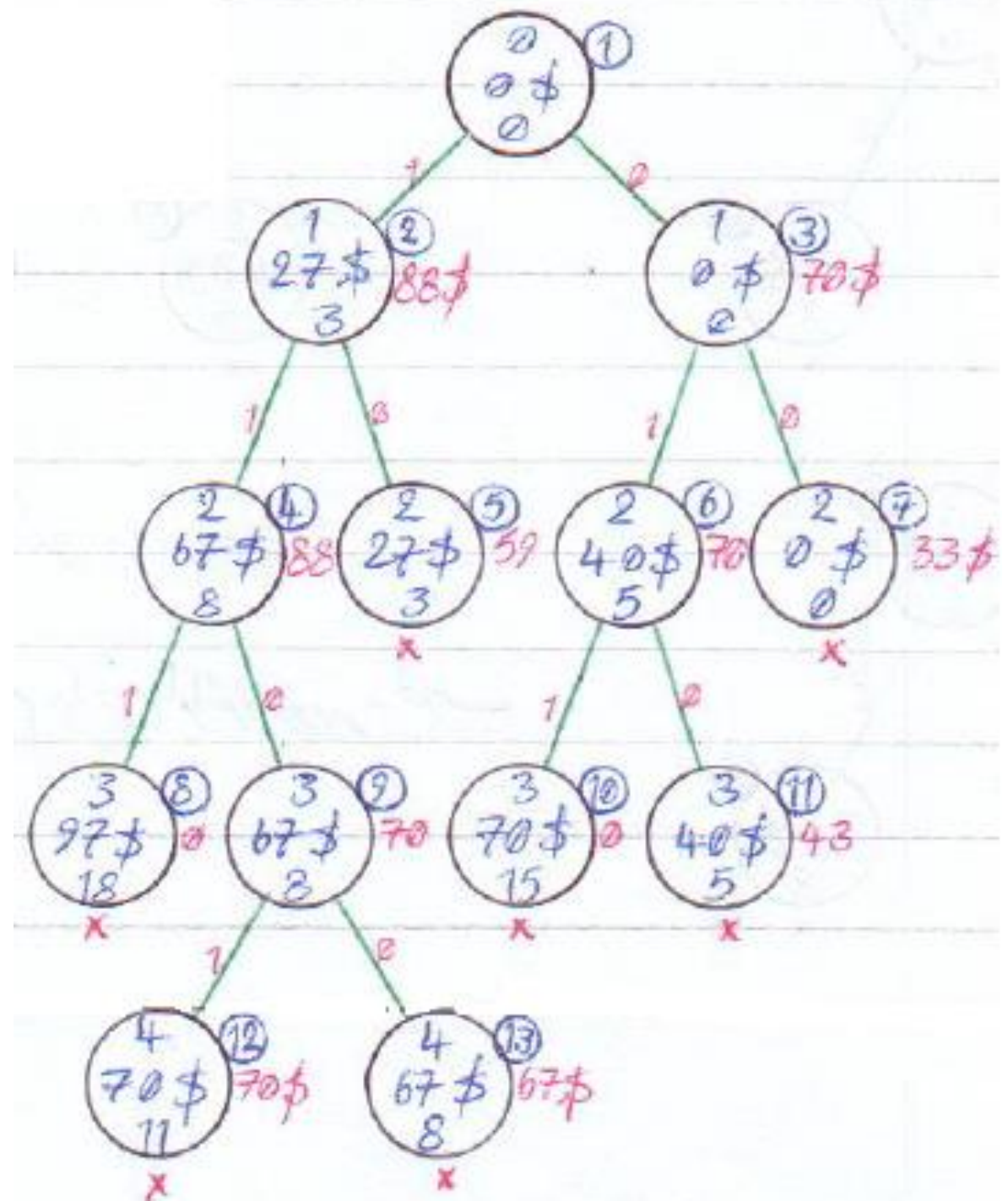
}

i	p_i	w_i
1	27\$	3
2	40\$	5
3	30\$	10
4	3\$	3

$$W = 15$$

هدف:

Maxprofit
0\$



Knapsack_BB_BF ($n, w[1..n], p[1..n], W$) {

Initialize (Q); Maxprofit = 0;

$v.i = v.profit = v.weight = 0$;

Enqueue (Q, v);

while not Empty (Q)

{

$v = \text{Dequeue}(Q)$;

$u.i = v.i + 1$, $u.profit = v.profit + p[v.i + 1]$,

$u.weight = v.weight + w[v.i + 1]$;

if $u.weight \leq w$ and $u.profit > \text{Maxprofit}$

Maxprofit = $u.profit$;

if Bound(u) > Maxprofit

Enqueue (Q, u);

$u.i = v.i + 1$, $u.profit = v.profit$, $u.weight = v.weight$;

if Bound(u) > Maxprofit

Enqueue (Q, u);

}

return Maxprofit;

Bound (v) {

if $v.\text{weight} \geq W$ return 0;

$k = v.i + 1$, $b = v.\text{profit}$, $\text{total} = v.\text{weight}$;

while $k \leq n$ and $\text{total} + w[k] \leq W$
{

$\text{total} += w[k]$;

$b += p[k]$;

$k++$;

}

if $k \leq n$

$b += [(W - \text{total}) / w[k]] \times p[k]$;

return b;

}

Bound (v) {

if $v.\text{weight} \geq W$ return 0;

$k = v.i + 1$, $b = \text{profit}$, $\text{total} = v.\text{weight}$;

while $k \leq n$ and $\text{total} + w[k] \leq W$
{

$\text{total} += w[k]$;

$b += p[k]$;

$k++$;

}

if $k \leq n$

$b += [(W - \text{total}) / w[k]] \times p[k]$;

return b;

}

→ while loop
is not executed
⇒ $O(n)$

• زمان اجرای کلی:

$$(1 + 2 + 2^2 + \dots + 2^n) \times n = (2^{n+1} - 1) \times n \Rightarrow \in O(n \times 2^n)$$

تمرین

- الگوریتم کوله پشتی ۱-۰ را برای ۵ شیء به روش شاخه و حد به صورت سطحی اجرا نمایید.
(درخت فضای حالت را ترسیم نمایید.)

شاخه و حد با بهترین حد (Best Bound)

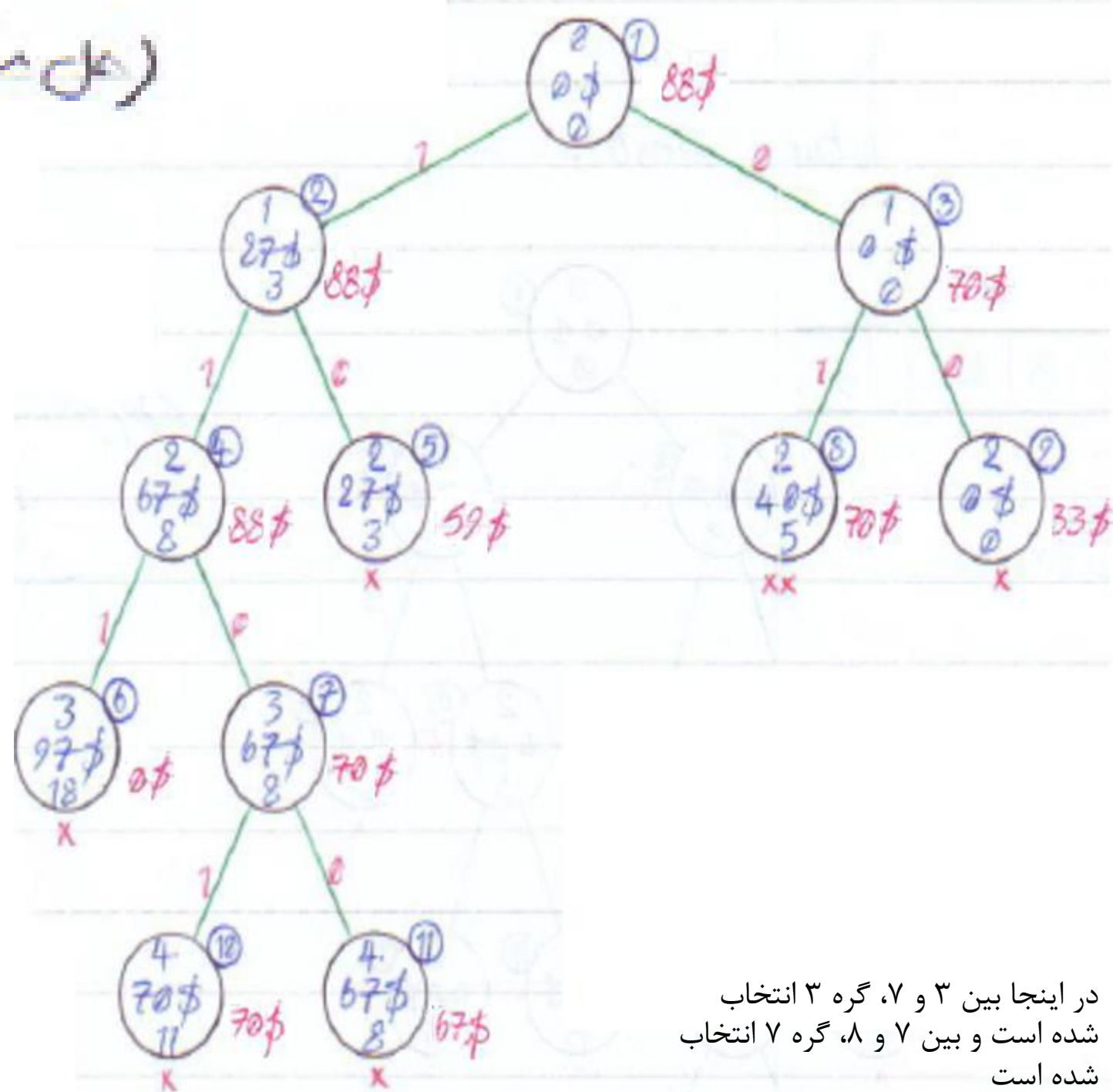
(Best Bound به معنی کره تری)

i	p_i	w_i
1	27\$	3
2	40\$	5
3	30\$	10
4	3\$	3

$$W = 15$$

صفت:

Maxprofit 0\$



Knapsack - BB-BF ($n, W[1..n], p[1..n], W$) {

 PInitialize (Q) ; MaxProfit = 0 ;

$v.i = v.profit = v.weight = 0$;

$v.bd = Bound(v)$;

 PEnqueue (Q, v) ;

 while not Empty (Q) .

 { $v = PDequeue(Q)$;

 if $v.bd > MaxProfit$

 { $u.i = v.i + 1$, $u.profit = v.profit + p[v.i + 1]$
 , $u.weight = v.weight + W[v.i + 1]$;

 if $u.weight \leq W$ and $u.profit > MaxProfit$

 MaxProfit = $u.profit$;

الگوریتم کوله پشتی ۱-۰
به روش شاخه و حد با بهترین حد

$u.bd = \text{Bound}(u);$

if $\underline{u.bd} > \text{Maxprofit}$

$\text{PEnqueue}(Q, u);$

$u.i = v.i + 1, u.profit = v.profit$

$, u.weight = v.weight;$

$u.bd = \text{Bound}(u);$

if $\underline{u.bd} > \text{Maxprofit}$

$\text{PEnqueue}(Q, u);$

} }

return Maxprofit;

}

الگوریتم کوله پشتی ۱-۰

به روش شاخه و حد با بهترین حد (ادامه)

• زمان اجرای کلی:

$$(1 + 2 + 2^2 + \dots + 2^n) \times n = (2^{n+1} - 1) \times n \Rightarrow \in O(n \times 2^n)$$

تمرین

- الگوریتم کوله پشتی ۱-۰ را برای ۵ شیء به روش شاخه و حد با بهترین حد اجرا نمایید.
(درخت فضای حالت را ترسیم نمایید.)

پایان