

Algorithm:

```

1. BB-Knapsack (cp, cw, K)
2.   cp ← 0 // Curr profit
3.   cw ← 0 // curr weight
4.   K ← 1 // Index
5.   if cw + w[K] ≤ M do
6.     Y[K] ← 1
7.     if K ≤ n do
8.       | BB-Knapsack (cp + P[K], cw + w[K], K+1)
9.       if cp + P[K] > fp and K = n
10.        | fp ← cp + P[K]
11.        | fw ← cw + w[K]
12.        | X ← Y
13.   if Bound (cp, cw, K) > fp do
14.     Y[K] ← 0
15.     if K ≤ n do
16.       | BB-Knapsack (cp, cw, K+1)
17.     if cp > fp and K = n
18.       | fp ← cp
19.       | fw ← cw
20.       | X ← Y
21.
22.

```

```

1. Bound (cp, cw, K)
2.   b ← cp
3.   c ← cw
4.   for i in (K+1, n) do
5.     | if c + w[i] ≤ M do
6.       | | c ← c + w[i]
7.       | | b ← b + P[i]
8.   return b

```

2

هر آستیک یک حداقل تعداد دارد. فرضی کنیم برای مالکسیم تبع به تعداد حداقل بیشتر نداریم. پس وزن

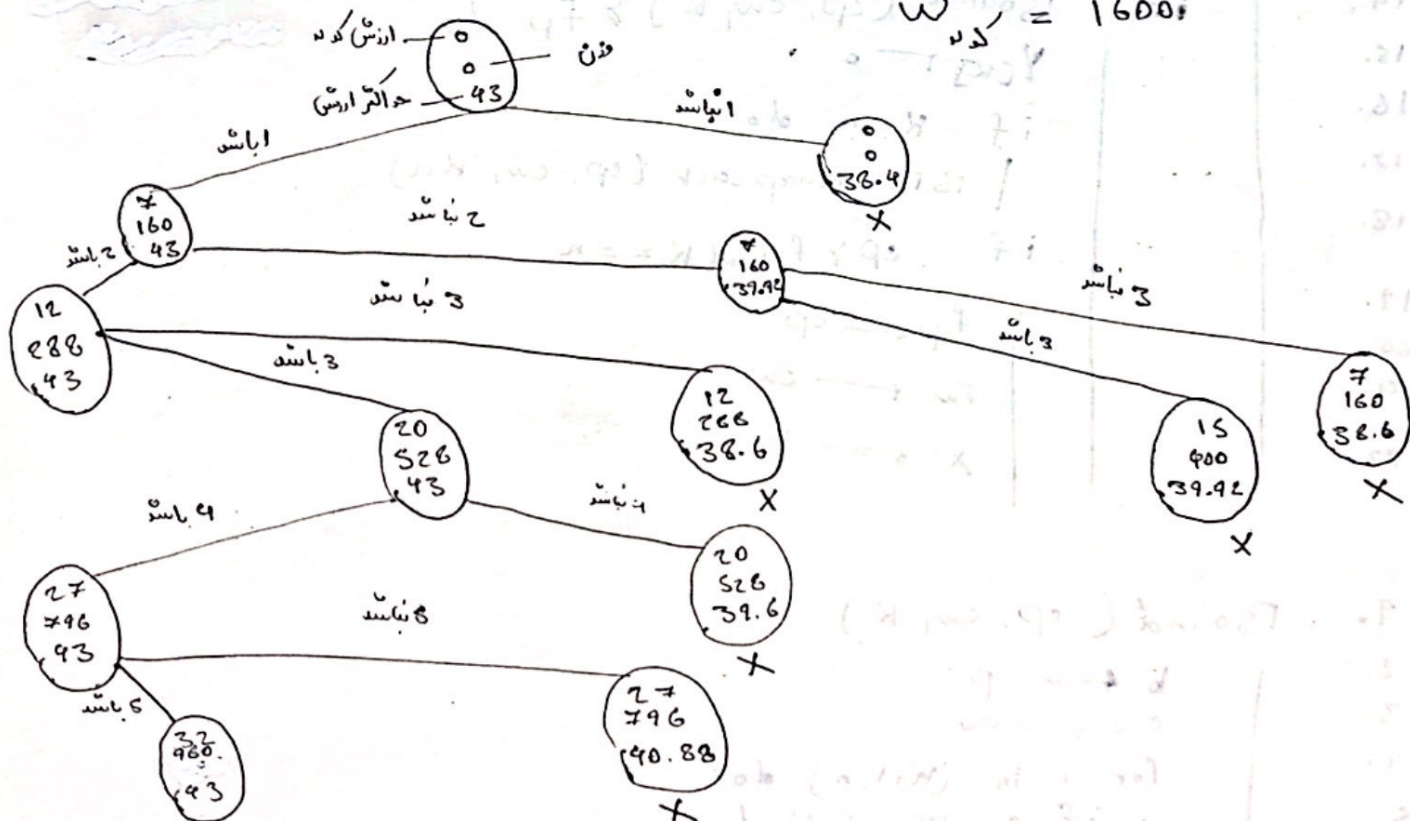
هر آستیک از جدول زیر می‌گردد: $W_p = \text{cost}$
 $\Rightarrow W_p = \text{cost}$ جدول جدید:

i	V_i	W_i	V_i/W_i
M	8	128	.039 2
C	7	240	.029 4
Y	3	160	.018 6
O	1	248	.004 10
R	8	240	.033 3 \Rightarrow
N	2	192	.010 9
B	7	160	.043 1
E	5	192	.026 5
WM	8	480	.016 7
RM	6	400	.015 8

$7 + 5 + 6 + 7 + 5 + 3 + 8 = 43$

i	V_i	W_i	V_i/W_i
1	7	160	43
2	8	128	39
3	8	240	33
4	7	240	29
5	6	192	26
6	3	160	18
7	8	480	16
8	6	400	15
9	2	192	10
10	1	248	4

$W_{\text{جدید}} = 1600$

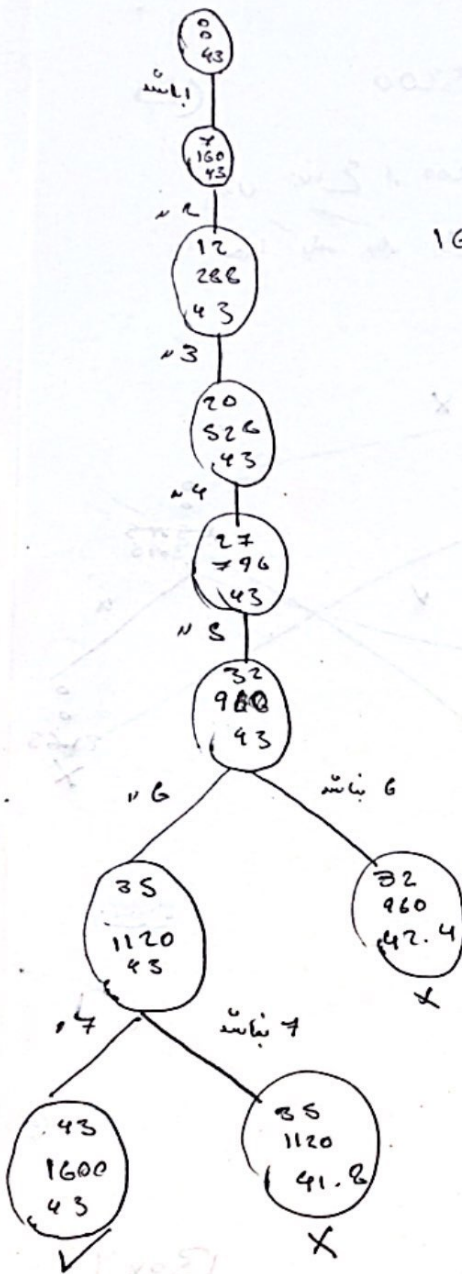


ادامه صفحه بعدی

الف) باید سداد ای 7 را برداریم.
چرا نقطه 1 برنج داریم؟ چون جمع دیتابیس کسی برابر 1600 شد.

items :

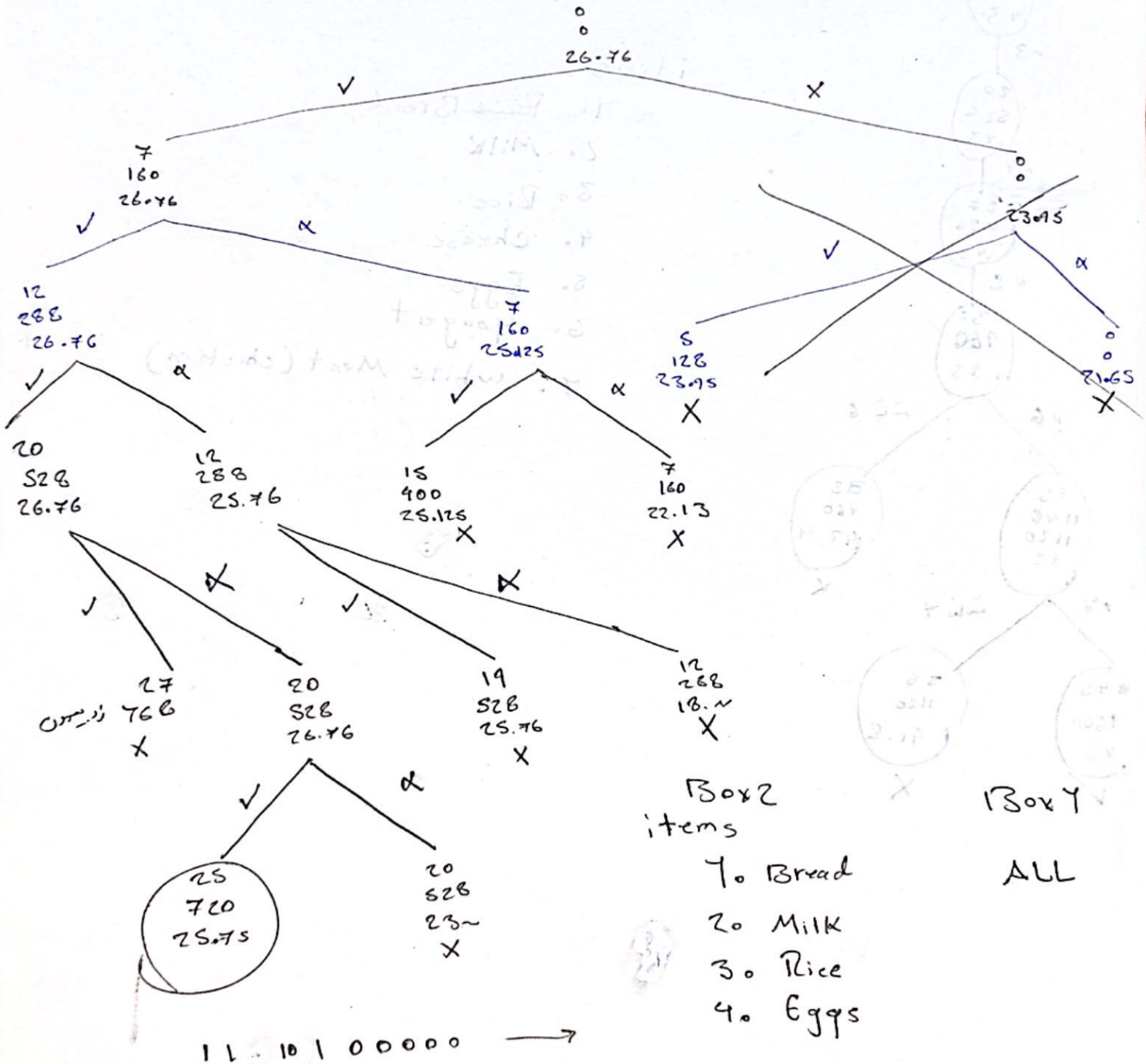
1. Rice Bread
2. Milk
3. Rice
4. Cheese
5. Eggs
6. Yougert
7. white Meat (chicken)



$$W' = 3200$$

(ب)

چون بهای از 3200 کمتر است هر محصول را یک بار می‌توان خرید. و به خرید هر یک 60 باقی می‌ماند.
 (یک جبهه از هر محصول یک جبهه دیگر داریم.)
 حالا باید دید برای 760 چقدر می‌توانیم. جواب سببی = $7 + 5 + 16 + 6 \cdot 76 = 26 \cdot 76$

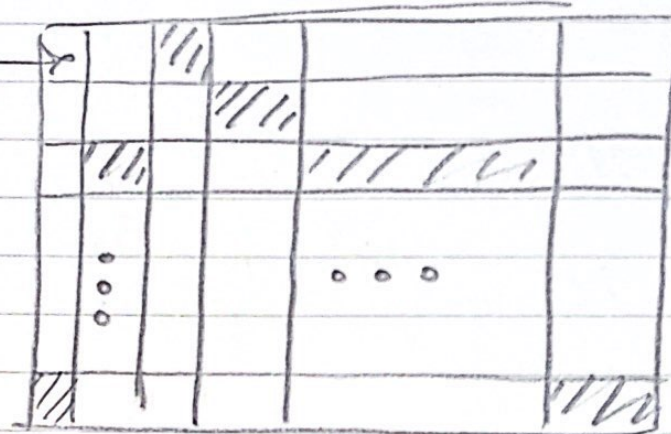


$P_i = (x, y)$ pairs.

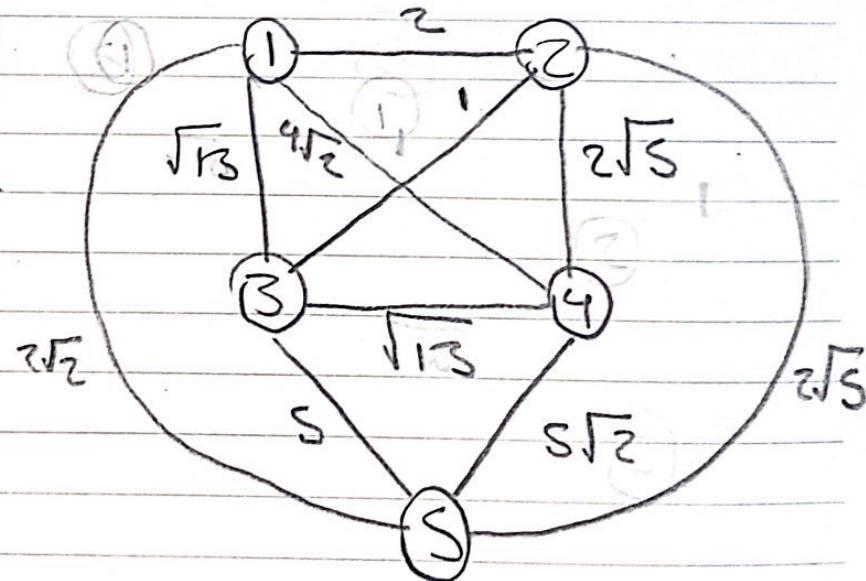
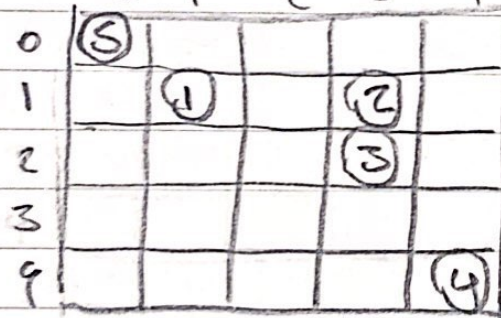
(3)

$1 \leq i \leq n$

start →



الر $n = 4$ ، D_i : دایره این شکل :
 $(1, 1)$ $(1, 3)$ $(2, 3)$ $(4, 4)$



چون 2 ماشین داریم می‌توانیم هوشمندانه عمل کنیم.

① به کمک الگوریتم کارگر یا کاربر استایلین یک

minimum cut انجام می‌دهیم.

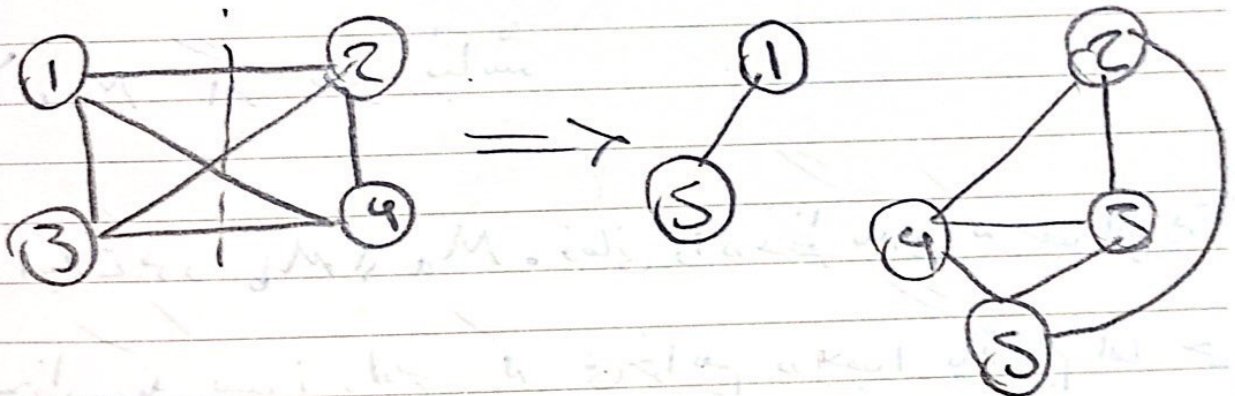
② که همان (5, 5) است و به هر 2 گران جدا شده

اضافه می‌کنیم (در مرحله قبل گران را 2 زیر گران تقسیم می‌کنیم)

③ به کمک Prim برای هر دو زیر گران یک MST می‌سازیم

④ هر ماشین روی MST زیر گران خود حرکت می‌کند.

⑤ این الگوریتم بهترین هزینه را خوبی می‌دهد.



(۹)

ابتداءً اول راه همان ترتیبی می‌خیم. سپس یک جعبه بری داریم و لیست ۱، ۲، ۳ تا جایی که در جعبه جا شود را قرار می‌دهیم. بعد جعبه دوم را بری داریم و الی آخر. این روش به این صورت است که اگر لیست نام در جعبه جا نشد یک جعبه جدید برداشته در آن قرار می‌دهیم.

تعداد جعبه‌ها = M_a

تعداد جعبه‌ها در بسته بنهایی لیست M_b
 $M_a \leq 2M_b$

(۱) $M_a = M_b$ می‌تواند درست باشد اگر مابقی‌ترین حالت را انجام دهیم

(۲) $M_a \geq 2M_b$ اگر بعینه نباشد

می‌توان اثبات کرد $M_a \leq 2M_b$. زمانی را در نظر بگیرید که جعبه اول پر می‌شود

در این لحظه یک لیست داریم که می‌خواهیم در جعبه‌ها بگذاریم اما حجم

فضای خالی جعبه کم‌تر از V_1 است. پس از آن جعبه دوم قرار می‌دهیم.

جعبه اول دوم روی هم می‌آید واحد حجم دارند و بیش از ۱ واحد شغل دارند.

اگر همین کار را برای جعبه ۳، ۴، ۵، ۶، الی آخر انجام دهیم به این

نتیجه می‌رسیم که $\frac{M_a}{2} \leq M_b$ می‌شود.