- Intro
- Fractional knapsak
- Free Cars

- Candy
- Max Jobs

Greedy Algorithm (short sightet in a positive way!)

Choose the best at current

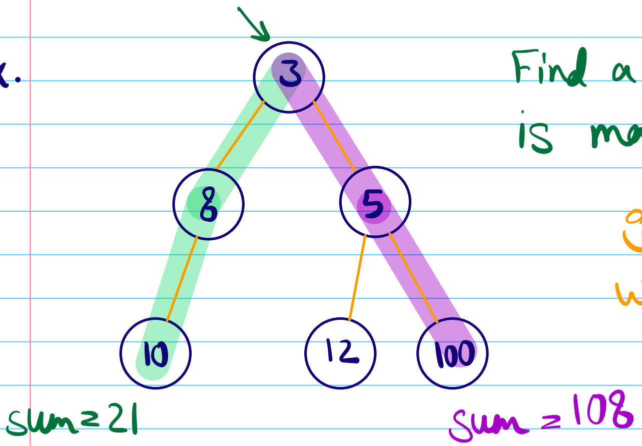
iphone price :

Amazon	$950
Apple	$999.99
eBay	$989.99

greedy

iphone announcement event

ex.



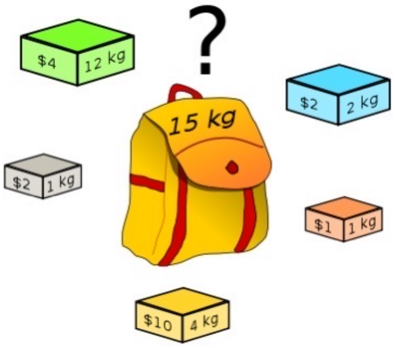
ans =0

Find a path that sum(nodes) is max

greedy aproach doesn't work all the time. Sometime it does.

max $

P1 Fractional knapsack8



protein in grams

Food	protein Content in the current amount			Protein / kg
Tomato	200	for	20 kg	10
Apple	180	for	15kg	12
Onion	250	for	50 kg	5
Chicken	150	for	10 kg	15
Potato	200	for	25kg	8
Mango	132	for	12kg	11
Seafood	100	for	5kg	20

16kg

1kg
20 → 5kg
6kg

maximiz protein in take if bag Capacity is 7 kg.

C#

```
double MaxProtein( int wt[], int protein[], int k ){
    List<(int, int, double)> a = new ...
    for (i=0; i< wt.Len; i++){
        p = (wt[i], protein[i], Protein[i]x1.0 / wt[i])
        a.Add(p)
    }
    Array.Sort(a, (t1, t2) => t1.density.cmp(t2.density) )
    double ans=0
    for(i=0; i< a.Len; i++){
        (w, p, d ) = a[i]
        if( w <= k){ ans += p; k=k-w }
        else{ ans+ =d*k ; break; }
    }
    ret ans
}
```

density

n

tc: O(nlogn)
SC: O(n)

ans ← 100

n

10 kg ,150, 15

10kg 150
2kg ?

2kg

Seefoo → 5kg

0-1 → what is? Futur session
Knapsack problem → greedy gives wrong answer

Free cars!

Car

P2 There is a limited time sale on toys. Sale is 100% off!! Free

$A[i]$ → sale end time for ith toy

$B[i]$ → Beauty of ith toy.

time starts from $t=0$, it takes 1 unit of time to buy one toy & toy can only be bought if $t < A[i]$.
→current time

buy toys s.t. sum of beauty of toys is maximized.

deal expiration time

ex $A = \{3, 1, 3, 2, 3\}$

beauty

$B = \{6, 5, 3, 1, 9\}$
1 2 3 4 5

t	toy#	b
0	5	9
1	1	6
2	3	3
① 3		18

T	toy#	b
0	2	5
1	5	9
2	1	6
3		
②		20

Quiz

$A = \{1 \quad 2\}$

$B = \{3 \quad 1500\}$
1 2

①

t	toy#	b
0	2	1500
1	x	

②

t	toy#	b
0	1	3
1	2	1500

1503

ex

	1	2	3	4	5	6	7	8
A = {	1	3	3	~~3~~	5	5	5	8
B = {	5	2	7	~~1~~	4	3	8	1

time stamp

b

t	A end time	B
0	1	5
1	~~3~~5	~~2~~8
2	3	7
3	5	4
4	5	3
5	5	8

Pair & Custom cmp

Code

A&B are sorted based on time ascending O(nlogn)

```
total Beauty=0   ; t=0 ;
for( i =0 ; i<n ; i++ }
{   if( t >=A[i]){
        if((B[i] > root of minheap)){
            b = minHeap.RemoveMin() //pop
            total Beauty -=b
            total Beauty+=B[i]
        } minHeap.insert(B[i])
    }else{   t< A[i]
        minHeap.insert(B[i])
        t++
        total Beauty+=B[i]
    }
}
```

check heap is not empty

log(n)

regret/ redeem

n

min heap

2,

2

5, 8, 7, 4, 3

TC: O(nlogn)

SC: O(n)

	1	2	3	4	5	6	7	8
A=	1	~~3~~	3	~~3~~	5	5	5	8
B=	5	~~2~~	7	~~1~~	4	3	8	1

↑ i

t= ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

B= ~~0~~ ~~5~~ ~~7~~ ~~14~~ ~~18~~ ~~21~~ ~~17~~ 27

(grades)

P3 There are N students with marks, teacher has to give them candies s.t.

a) Every student shand have at least one candy.

b) Students with more marks than neighbors have more candies than them. ↘ immediate neighbors

Find min candies to distribute.

ex

$$7 \leftarrow A = \{\overset{0}{\underset{1}{1}}, \overset{1}{\underset{3}{5}}, \overset{2}{\underset{2}{2}}, \overset{3}{\underset{1}{1}}\}$$

Quiz $\boxed{A = \{\overset{0}{\underset{1}{4}}, \overset{1}{\underset{1}{4}}, \overset{2}{\underset{1}{4}}, \overset{3}{\underset{1}{4}}, \overset{4}{\underset{1}{4}}\}}$ ans= 5

$$7 \leftarrow \text{ex } A = \{\overset{0}{\underset{1}{8}}, \overset{1}{\underset{3}{10}}, \overset{2}{\underset{2}{6}}, \overset{3}{\underset{1}{2}}\}$$

i-1 i Pass 1 <

Pass 2 >

Quiz $\boxed{A = \{\underset{1}{1}, \underset{~~1~~2}{6}, \underset{~~1~~2}{3}, \underset{1}{1}, \underset{~~1~~2}{10}, \underset{~~1~~3}{12}, \underset{~~1~~4}{20}, \underset{~~1~~2}{5}, \underset{1}{2}\}}$ ans= 19

c[ ]

3

① for all i, c[i] = 1

② for all i, if $(\underbrace{A[i-1] < A[i]}_{\text{Pass 1}})$ ⇒ if(c[i-1] >= c[i]) c[i] = c[i-1]+1

for i = 1 → n-1

grade

Pass 1 <

TC: O(n)

SC: O(n)

③ for all i, if $(\underbrace{A[i] > A[i+1]}_{\text{Pass 2}})$ ⇒ if(c[i] <= c[i+1]) c[i] = c[i+1]+1

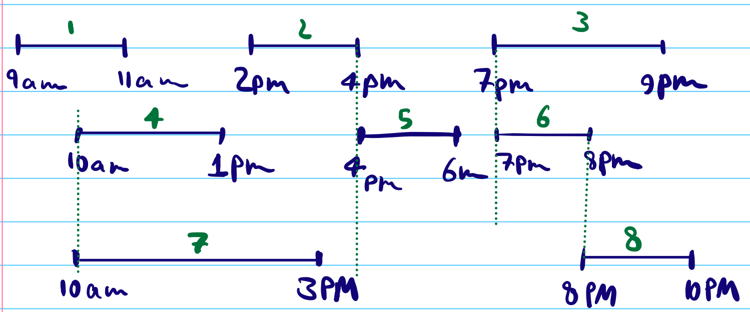
for i = n-2 → 0

Pass 2 >

Facebook

P4 Given n jobs, with their start & end time, find max jobs that can be completed if only one job can be done at a time.

ex



Quiz

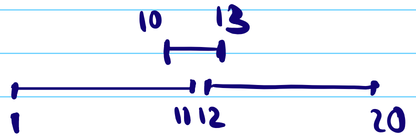
$S = \{ \overset{1}{1}, \overset{2}{5}, \overset{3}{8}, \overset{4}{7}, \overset{5}{13}, \overset{6}{12} \}$

$e = \{ \underbrace{2}_{1}, \underbrace{10, 10, 11}_{1}, \underbrace{19, 20}_{1} \}$

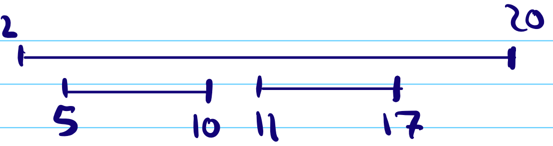
ans = 3

duration[i] = e[i] - s[i]

idea1 duration ✗



idea2 start time? ✗



idea3 end time → sort ascending

start early + less duration

Code

```
arr<(s,e)>    a = new...
  for (i=0 → n-1) a.add(s[i], e[i]))
    a.Sort( cmp )
    //sort based on end time

    ans = 1
    lastEnd = a[0].e
    for  i → 1 to (N-1){
            if(a[i].s >= lastEnd){
                  ans++;
                  lastEnd = a[i].e
            }
              //else NoOp
    }
    ret  ans;
```

```
        0   1   2   3   4    5
s = { 1,  5,  8,  7,  13,  12 }
e = { 2,  10, 10, 11, 19,  20 }
                      ↗
                  last
                  end
```

ans = 3