

coin change perm

coin change combination

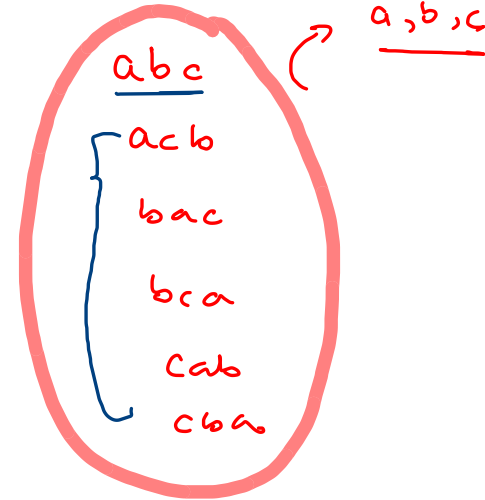
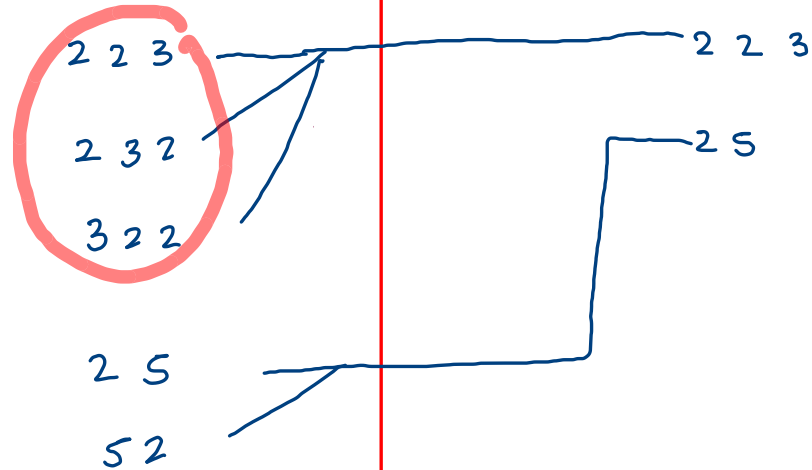
Arrangement

selection

2, 3, 5

amt = 7

str = abc



amt = 7, coins = [2, 3, 5]

ccp  
↙

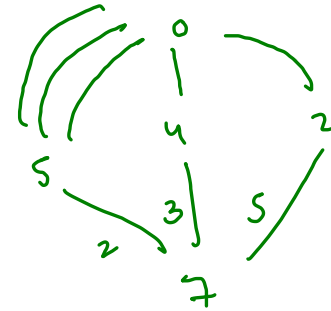
1	0	1	1	2	3	2	5
0	1	2	3	4	5	6	7

.		2.	3.	22.	32	222.	322. 223.
					23.	33.	232. 25.
					5.		52.

dp[i] → num to pay i amt.

dp  
coins



ccc  
↙

amt = 7,

coins = [2, 3, 5]

~~2~~ ~~3~~ ~~5~~

1	0	1	1	1	2	1	2
0	1	2	3	4	5	6	7
-		2.	3.	22.	23. 5.	222. 33.	223. 25

coins  
dp

⑤ 3, 2

2      3.      5      6      7  
2.      3.      5.      33.      52.  
32.                     322.

53

35

53 ✓

35 ✗

tabs

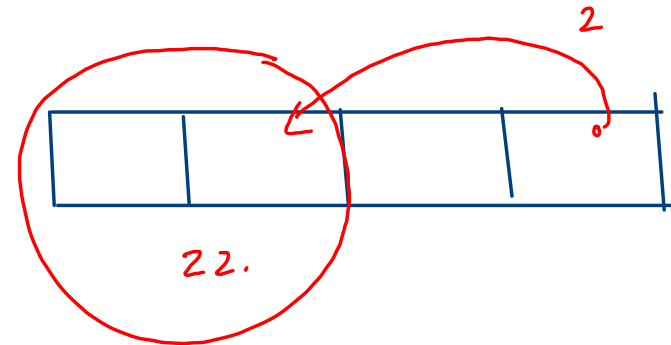
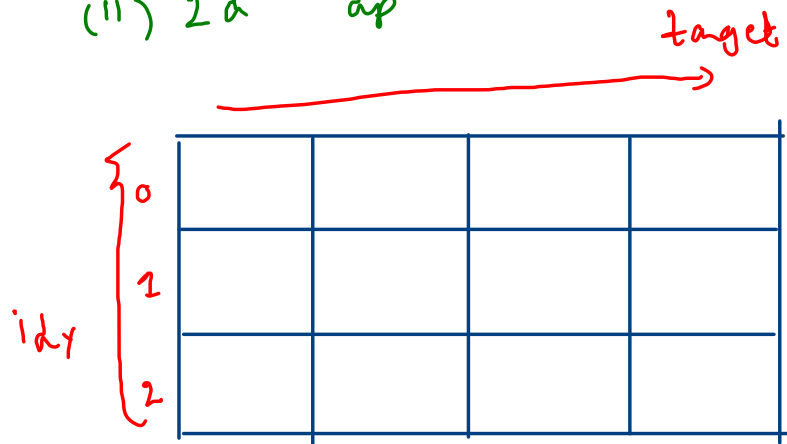
coin change

(i) no duplicacy

(i) duplicacy

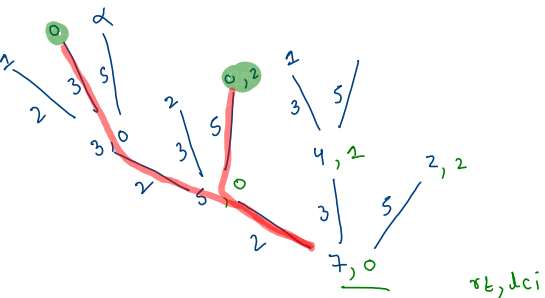
(ii) 2d dp

(ii) 1d dp



ccc1

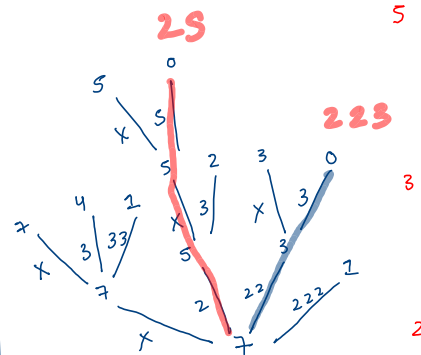
target, dci



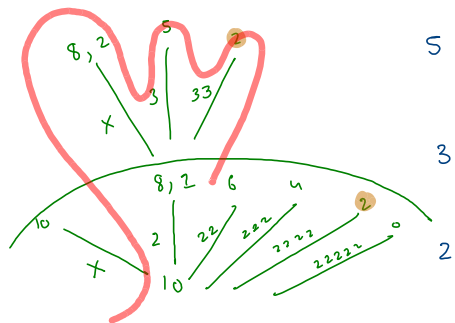
rk, dci

ccc2

2d



memoisation -> 2d  
labulation -> 1d



2,3

	0	1	2	3	4	5
0-2	✓	✗	✓			
1-3	✓	✗	✗	✓	✓	✗
2-4	✓	✗	✗	✗	✓	✗

target = 5

den = 2,3,4

dp[i][j] ->

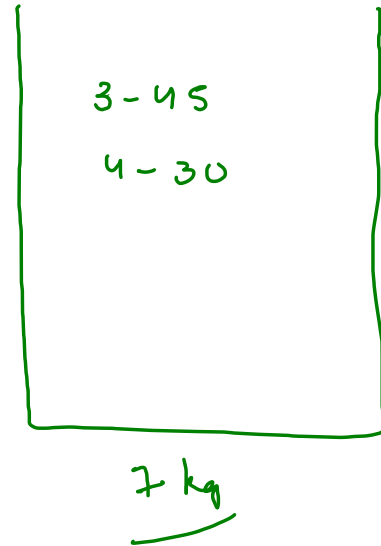
5  
15 14 10 45 30  
2 5 1 3 4  
7

	0	1	2	3	4
val $\rightarrow$	15	14	10	45	30
wt $\rightarrow$	2	5	1	3	4

Longest sum

(i) un-breakable

(ii) one item can be  
used once.



cap  $\rightarrow$  7

val  $\rightarrow$  15 14 10 45 30

wt  $\rightarrow$  2 5 2 3 4

	0	1	2	3	4	5	6	7
2-15	0	0	15	15	15	15	15	15
5-14	0	0	15	15	15	15	15	29
1-10	0	10	15	25	25	25	25	29
3-45	0	10	15	45	55	60	70	70
4-30	0	10	15	45	55	60	70	70

```

else {
    int exc = dp[i-1][j];
    int inc = 0;

    if(j >= wt[i]) {
        inc = dp[i-1][j-wt[i]] + val[i];
    }

    dp[i][j] = Math.max(exc, inc);
}

```

exc = 25

inc = 10 + 45

$dp[i-1][j-wt[i]]$     $dp[i-1][j]$   
 inc   exc  
 $dp[i][j]$

```

public static int ZOKnapsack(int[] val, int[] wt, int cap) {
    int[][] dp = new int[val.length][cap+1];

    for(int i = 0; i < dp.length; i++) {
        for(int j = 0; j < dp[0].length; j++) {
            if(i == 0) {
                //there is only one item
                if(j >= wt[i]) {
                    dp[i][j] = val[i];
                }
                else {
                    dp[i][j] = 0;
                }
            }
            else if(j == 0) {
                //cap is zero
                dp[i][j] = 0;
            }
            else {
                int exc = dp[i-1][j];
                int inc = 0;

                if(j >= wt[i]) {
                    inc = dp[i-1][j-wt[i]] + val[i];
                }

                dp[i][j] = Math.max(exc, inc);
            }
        }
    }
}

```

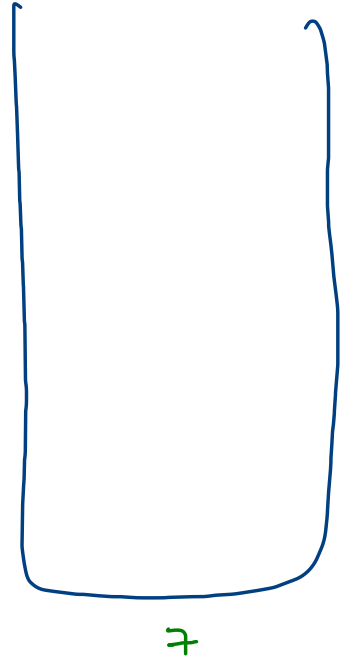
cap  $\rightarrow$  7

	0	1	2	3	4
val $\rightarrow$	15	14	10	45	30
wt $\rightarrow$	2	5	2	3	4

(i) un-breakable

(ii) infinite supply of  
every item.

(iii) coin change comb or  
perm both are valid.





0      1      2      3      4  
 val -> 15    14    10    45    30  
 wt -> 2      5      2      3      4

~~max = 35~~  
 55

2, 20+15  
 X  
 3, 55  
 1, 10+45  
 0, 0+30

```

for(int i = 1; i <= cap; i++) {
    int max = 0;
    for(int j=0; j < val.length; j++) {
        if(i >= wt[j]) {
            int rem_cap = i - wt[j];
            int pr = dp[rem_cap] + val[j];

            if(pr > max) {
                max = pr;
            }
        }
    }
    dp[i] = max;
}
  
```

0	10	20	45	55	65	90	100
0	1	2	3	4	5	6	7
	1	11	3	31	311	33	331

~~000~~

~~001~~

010 ✓

011 ✓

~~100~~

101 ✓

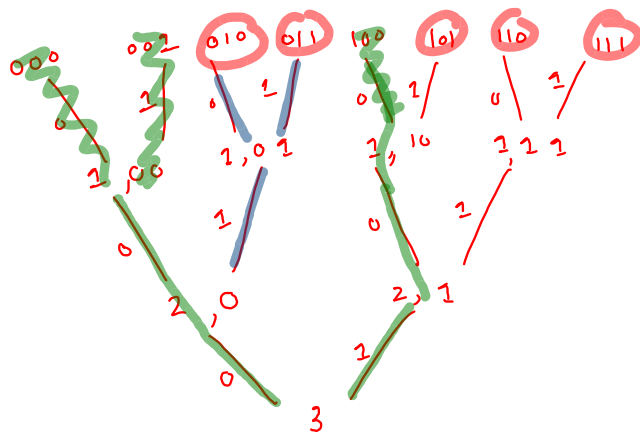
110 ✓

111 ✓

$n=3$

ans = 5

	0	1	2	3	4
ending at 0	0	1 $\begin{array}{ c } \hline 0 \\ \hline \end{array}$	1 $\begin{array}{ c } \hline 10 \\ \hline \end{array}$	2 $\begin{array}{ c } \hline 010 \\ 110 \\ \hline \end{array}$	3
ending at 1	0	1 $\begin{array}{ c } \hline 1 \\ \hline \end{array}$	2 $\begin{array}{ c } \hline 01 \\ 11 \\ \hline \end{array}$	3 $\begin{array}{ c } \hline 101, 111 \\ 011 \\ \hline \end{array}$	5



ans = 3 + 5

$n = 5$

```
public static int cbs(int n) {  
    int oetz = 1;  
    int oeto = 1;  
  
    for(int i = 2; i <= n; i++) {  
        int netz = oeto;  
        int neto = oetz + oeto;  
  
        oeto = neto;  
        oetz = netz;  
    }  
  
    return oeto + oetz;  
}
```

	0	1	2	3	4	5
ending at 0	0	1	1	2	3	<sup>02</sup> 5
ending at 1	0	1	2	3	5	<sup>00</sup> 8

Ans = 13

5

