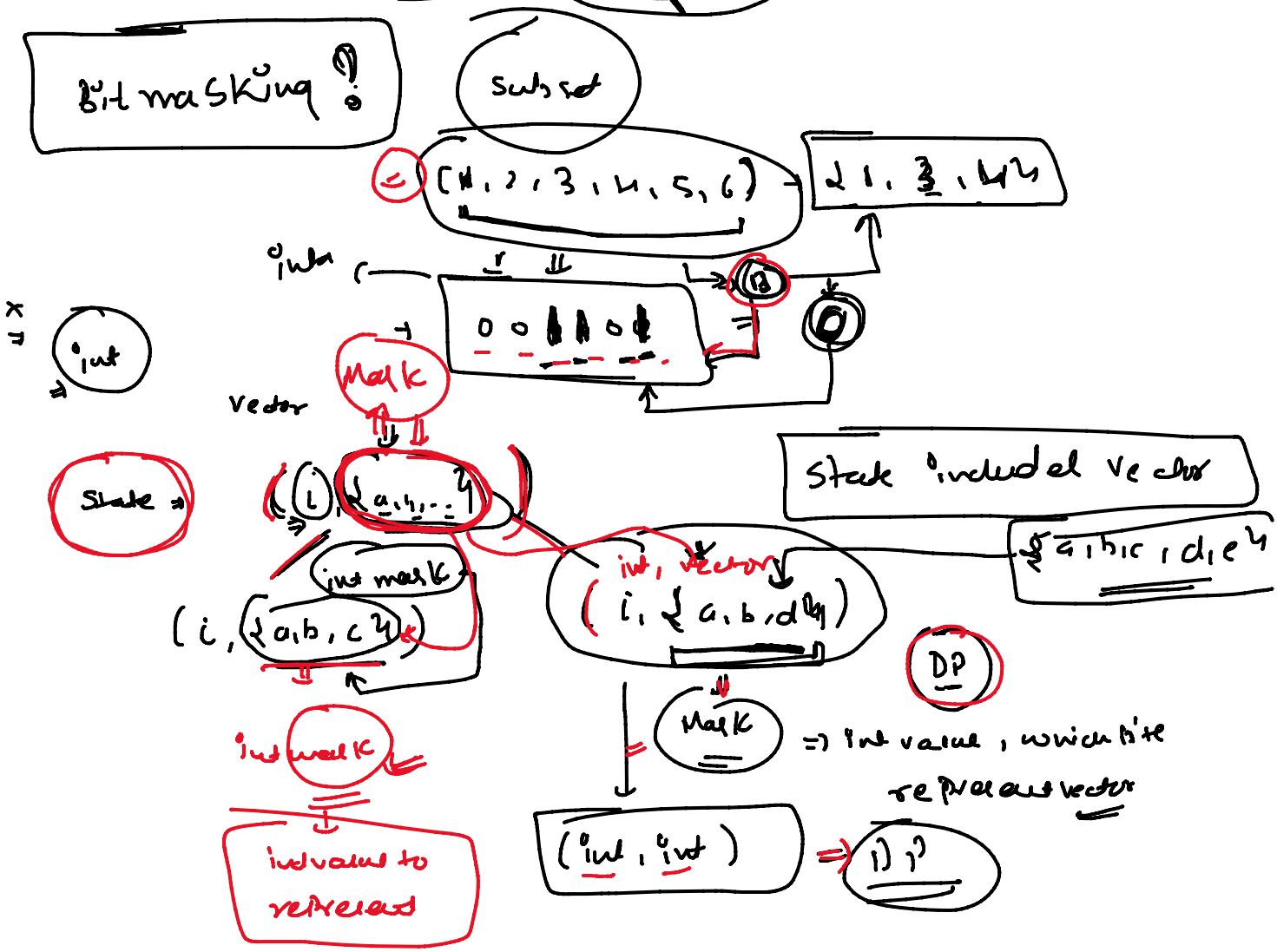
Class 91

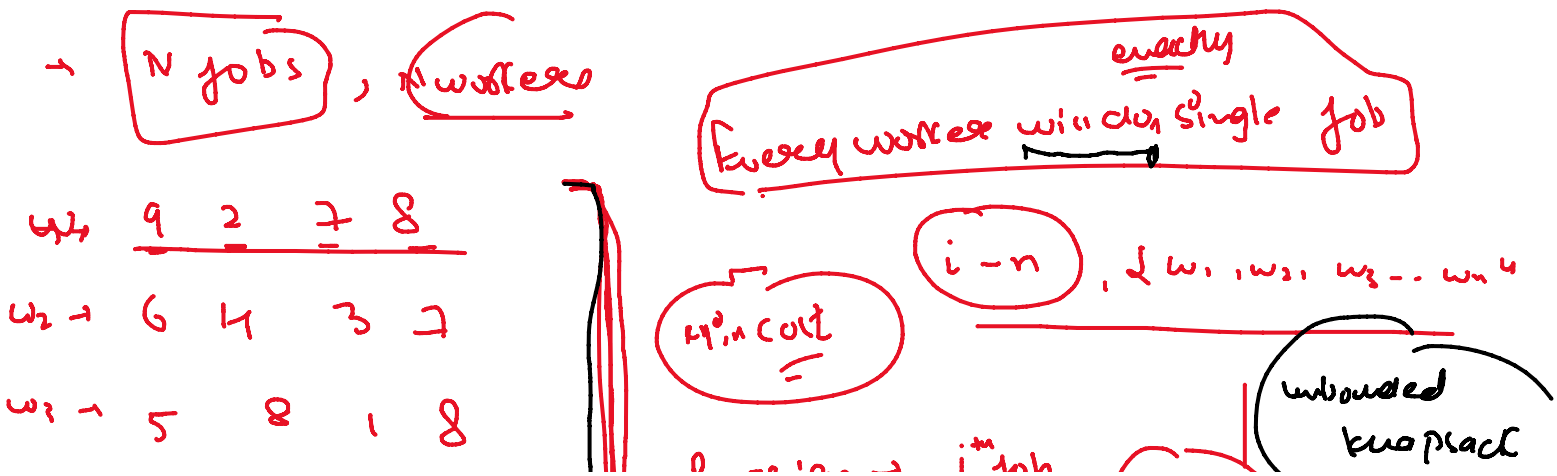
22 May 2023 21:08

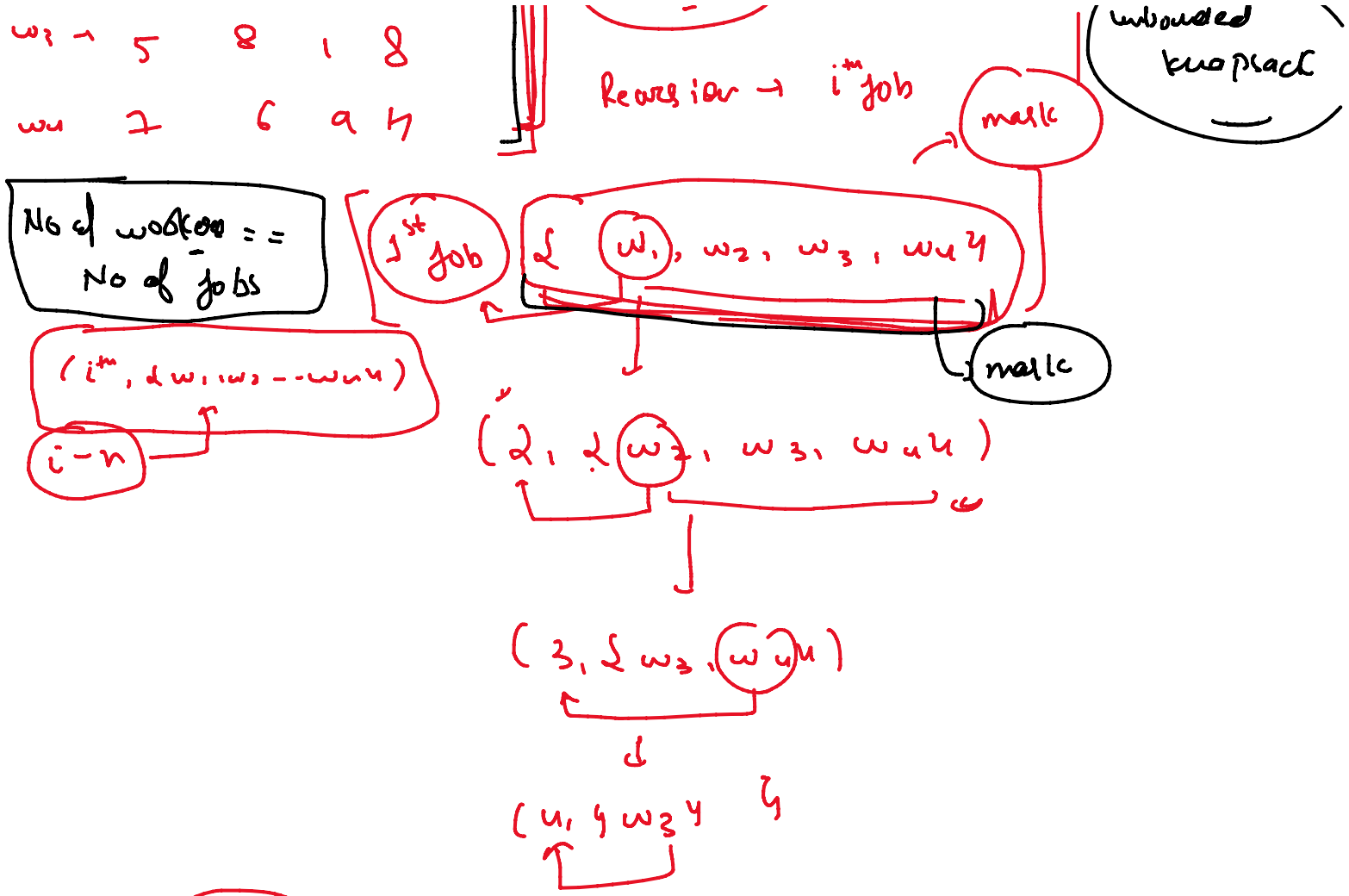
Class 91

Dynamic Programming

DP with Bitmasking (Adv)








```
int solve ( int i, int mask, vector<vector<int>> workers)
{
    int n = workers.size();
    if (i == n)
        return 0;
    if (dp[i][mask] != -1) return dp[i][mask];
    int ans = INT_MAX;
    for (int j = 0; j < n; j++)
```

n workers

visited

0 0 → 0 | 0 1 → 1 | 1 0 → 1 | 1 1 → 0

(1<<n) - 1

visited

for(int j = 0; j < n; j++)

if(!(mask & (1<<j)))

{ int cost = workers[j][i] + solve(i+1, mask | (1<<j), workers);

ans = min(ans, cost);

}

dp[i][mask] =

return ans;

(1<<n) - 1

0 0 1 1 → mask

Backtracking / Recursion

1

k Equal sum Subset

Mask

[4, 3, 2, 3, 5, 2, 1]

k = 4

20/4 = 5

total sum % k == 0

20/4 = 5

target = total sum / k

Sum required in Each Bucket !

int target = total sum / k ;

int n = nums.size();

int sum = 0;

```
vector<int> dp(1<<n, -1);
dp[0] = 0;  int sum = 0;
For (i=0; i<n; i++)
      sum += nums[i];
if (sum % k) return false;
int tar = sum / k;

For (int mask=0; mask < (1<<n); mask++)
{  For (i=0; i<n; i++)
   {  if ( !(mask & (1<<i)) && dp[mask] + nums[i] <= tar)
         dp[mask | (1<<i)] = (dp[mask] + nums[i]) % tar;
   }
}

return dp[(1<<n)-1] == 0;
```

4, 2, 3 = 9

5

dp[mask]

dp[mask] % tar

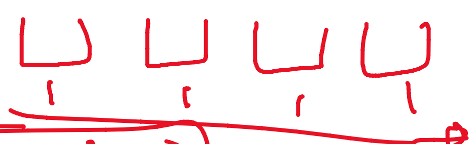
{1, 2, 3, 4}

dp[mask] = 4

5-1 = 4

0

dp[mask] =



bucket

dp[mask] == 0

same element

dp[mask] = 0

dp[mask] = 1

target = x

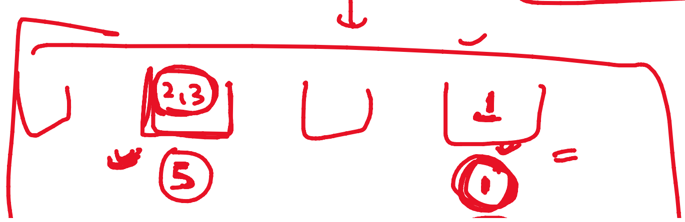
dp[mask] = 0

[2, 3, 5]

dp[mask] = 0

dp[mask] = 5

(2, 3)



(2, 3)

5

1 =

dp [mask] =

{2, 3}

mask = {1, 2, 3}

dp [mask] = 1

dp [mask] = 1

Can I win

A B

desiredTotal = 11

≤ 1

maxNo = 10

1 - 10

1 + 10 => 11 A

{2 - 10} {1, 10}

2 - 10

mask

```
bool win (int maxNO, int desTotal, int visited)
{ if (desTotal ≤ 0)
      return 0;

  int ans

  for (i = 1; i <= max; i++)
  { if ((1 << i) & visited == 0))
    {
      ans = win ( maxNo, des total - i, visited | (1 << i) )
```

1st player

2nd player loose

player 2

{ ans = win(maxv, del total - i, visited | (1<<i))

if (ans == 0)

return 1;

}

}

return 0;

}