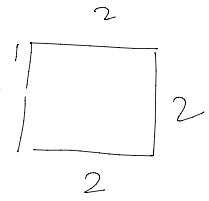
15 May 2024 21:34

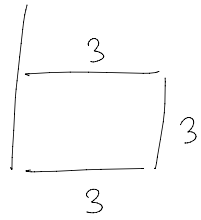
matchsticks to squares

[1,1,2,2,2]



true

[3,3,3,3,4]



false

```
bool makesquare(vector<int> matchsticks){
    bool ans=0;
    recur(ans,0,0,0,0,matchstick,0);
    return ans;
}

void recur(ans,s1,s2,s3,s4,MS,ind);
    if(ind==MS.size()){
        if(s1==s2 && s2==s3 && s3==s4){
            ans=true;
        }
    }
    return;
}
```

stopping condition

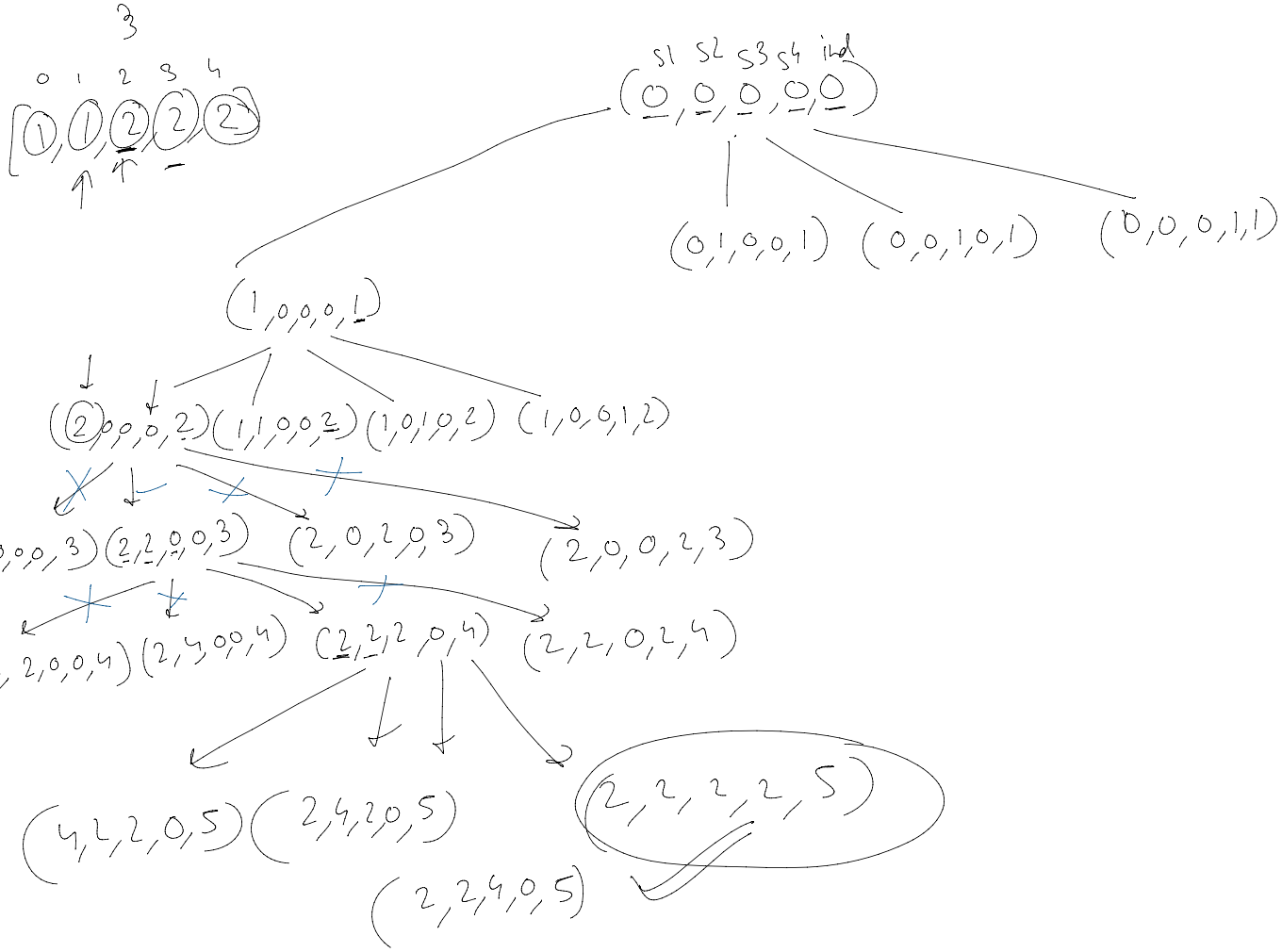
① recur(ans, s1+MS[ind], s2, s3, s4, MS, ind+1);

① recur(ans, s1+MS[ind], s2, s3, s4, MS, ind+1);

② recur(ans, s1, s2+MS[ind], s3, s4, MS, ind+1);

③ recur(ans, s1, s2, s3+MS[ind], s4, MS, ind+1);

④ recur(ans, s1, s2, s3, s4+MS[ind], MS, ind+1);



every matchstick has 4 option → s1, s2, s3, s4

n matchstick: 4^n

Pruning

Pruning

```
bool makesquare(vector<int> MS) {
    if(MS.size()<4) { return false; }
    int sum=0;
    for(int i=0; i<MS.size(); i++) {
        sum += MS[i];
    }
    if(sum%4!=0) { return false; }
    int sidelen = sum/4;
    bool ans = false;
    // sort MS in descending order.
    recur(ans, 0,0,0, sidelen, MS, 0);
    return ans;
}

void recur(ans, s1, s2, s3, sidelen, MS, ind) {
    if(ans==true) {
        return;
    }
    if(s1>sidelen || s2>sidelen || s3>sidelen) {
        return;
    }
```

②
```
return;
```

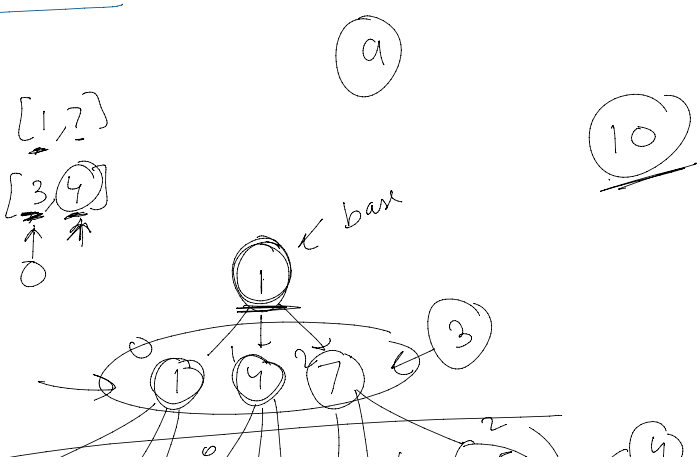
③
```
if(ind == MS.size()){
    if(s1==sidelen && s2==sidelen && s3==sidelen){
        ans=true;
    }
    return;
}
rec(ans, s1+MS[ind], s2, s3, sidelen, MS, ind+1);
rec(ans, s1, s2+MS[ind], s3, sidelen, MS, ind+1);
rec(ans, s1, s2, s3+MS[ind], sidelen, MS, ind+1);
rec(ans, s1, s2, s3, sidelen, MS, ind+1);
```

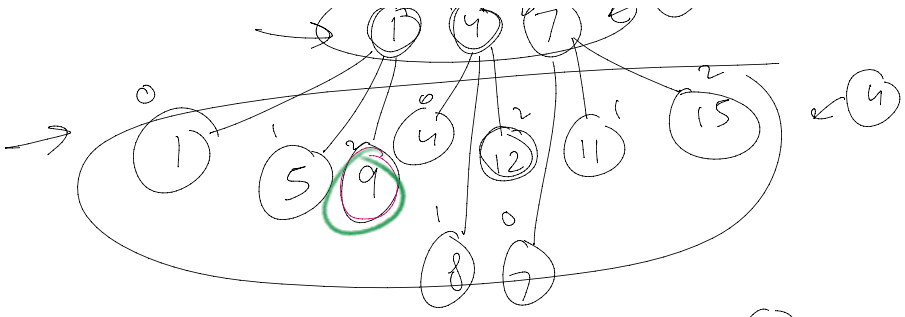
Closist Dessert Cost

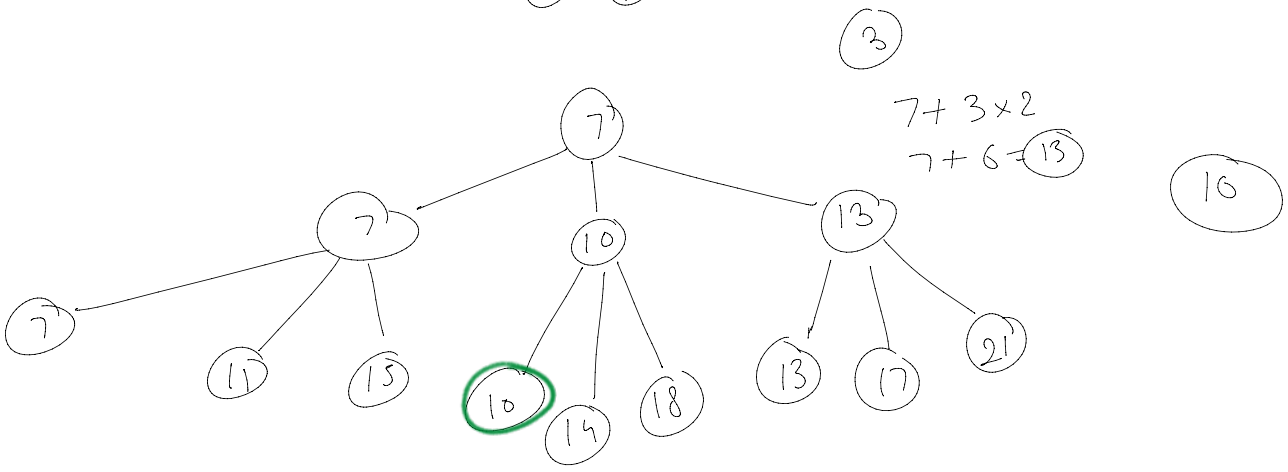
⇒ base cost = [1,7]

⇒ Topping cost = [3,4]

target = 10







```
int closestCost(BC, TC, Target) {
    int ans = INT_MIN;

    Loop through BC(i):
        recur(BC[i], TC, target, ans, 0);

3
recur(Total cost, TC, target, ans, ind) {
    if (abs(Total cost - target) < abs(ans - target)) {
        ans = Total cost;
    }
3
if (Total cost > target) {
```

```
}
① if( Total cost > target ){
        return;
   }
② if( ind == TC.size() ){
        return;
   }

recur( Total cost, TC, target, ans, ind+1 );
recur( Total cost + TC[ind], TC, target, ans, ind+1 );
recur( Total cost + 2*TC[ind], TC, target, ans, ind+1 );

}
```