1478. Allocate Mailboxes <Hard>

class Solution {

public:

int minDistance(vector<int>& houses, int k) {

int n = houses.size();

sort(houses.begin(),houses.end());

vector<vector<int>> dp(n,vector<int>(k+1,INT\_MAX));

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

dp[i][1] = findDist(houses,-1,i);

for(int j=2;j<=k;++j)

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

for(int h=0;h<i;++h)

dp[i][j] = min((double)(dp[i][j]),dp[h][j-1]+findDist(houses,h,i));

return dp[n-1][k];

}

double findDist(vector<int>& houses, int h, int i){

double mid, dist = 0;

if((i-h) % 2 == 1)

mid = houses[(i+h+1)/2];

else

mid = (houses[(i+h)/2] + houses[(i+h)/2+1])/(double)2;

for(int cur = h+1; cur <= i; ++cur)

dist += abs(mid-houses[cur]);

return dist;

}

};

1478. Allocate Mailboxes <Hard>

class Solution {

public:

int minDistance(vector<int>& houses, int k) {

int n = houses.size();

sort(houses.begin(), houses.end());

vector<vector<int>> dis(n, vector<int>(n));

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

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

dis[i][j] = getdis(houses, i, j);

vector<vector<int>> dp(k + 1, vector<int>(n, INT\_MAX));

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

dp[1][i] = dis[0][i];

for(int cnt = 2; cnt <= k; cnt ++)

for(int end = cnt - 1; end < n; end ++)

for(int start = end; start >= cnt - 1; start --)

if(dp[cnt - 1][start - 1] != INT\_MAX)

dp[cnt][end] = min(dp[cnt][end], dis[start][end] + dp[cnt - 1][start - 1]);

return dp[k][n - 1];

}

private:

int getdis(const vector<int>& houses, int start, int end){

int mid = (start + end) / 2;

int res = 0;

for(int i = start; i <= end; i ++) res += abs(houses[i] - houses[mid]);

return res;

}

};