class Solution {

public:

double mincostToHireWorkers(vector<int>& quality, vector<int>& wage, int k) {

ios\_base::sync\_with\_stdio(false);

cin.tie(nullptr);

cout.tie(nullptr);

vector<pair<double, int>> ratio;

int n = quality.size();

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

ratio.emplace\_back(static\_cast<double>(wage[i]) / quality[i], i);

}

sort(begin(ratio), end(ratio));

priority\_queue<int> maxHeap;

int qualitySum = 0;

double maxRate = 0.0;

for (int i = 0; i < k; ++i) {

qualitySum += quality[ratio[i].second];

maxRate = max(maxRate, ratio[i].first);

maxHeap.push(quality[ratio[i].second]);

}

double res = maxRate \* qualitySum;

for (int i = k; i < n; ++i) {

maxRate = max(maxRate, ratio[i].first);

qualitySum -= maxHeap.top();

maxHeap.pop();

qualitySum += quality[ratio[i].second];

maxHeap.push(quality[ratio[i].second]);

res = min(res, maxRate \* qualitySum);

}

return res;

}

};

class Solution {

public:

vector<long long> jugglerSequence(long long n) {

// code here

vector<long long> ans;

while(n!=1) {

ans.push\_back(n);

if(n%2==0)

n=sqrt(n);

else

n=n\*sqrt(n);

}

ans.push\_back(1);

return ans;

}

};

Link : <https://www.geeksforgeeks.org/problems/juggler-sequence3930/1>