* Chocolate Distribution problem :

m students.

* each gets 1 chocolate (mtlb total m required)

arr = [3, 4, 1, 9, 56, 7, 9, 12]

m = 5;

sort → (5) ese elements utane hai jo min/max small.

[1, 3, 4, 7, 9, 9, 12, 56]

→ min ... max ... max.

→ (8) [i-m] [m-i]

[7, 3, 2, 4, 9, 12, 56], m = 3.

[3, 2, 4, 7, 9, 12, 56]

(m).

{ To isme Fixed Window Sliding use kr sakta hai!!
→ phle sort karenge fir.
→ phle (m window k max-min)
→ fir age ko karna hai (calculate); } Approach

```
sort(a.begin(), a.end())
int dif = INT_MAX;
dif = min(dif, a[m-1], a[0]);      // phle window k calculate
for(int i=m; i<n; i++)
  { dif = min(dif, a[i] - a[i-m]);
  }
return dif //
```

Minimum cost of Ropes :

arr = [] ropes.

☆ Connect all to single, with minimum total cost

☆ cost of 2 ropes is sum of their length;

arr = [4, 3, 2, 6], n=4

sort → [2, 3, 4, 6] → 5

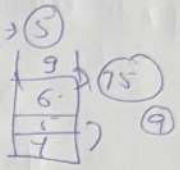
sort → [4, 5, 6] → 9

sort → [9, 6]

add → [15]

→ 5+9+15

→ (29)



[2, 3, 4, 6] V = [5 | 9]

vector[0] + arr[2]

vector

Alternative

→ min heap banao.

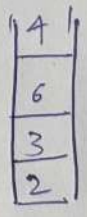
merge → (5).

→ add back to heap

```
int sum = 0;
priority_queue < int, vector< int >, greater< int >> q;
for(int i=0; i< arr.size(); i++)
    q.push(arr[i]);
vector<int>
while( q.size() > 1)
    int a = q.top();
    q.pop()
    int b = q.top()
    q.pop()
    q.push (a+b);
    sum = sum + a+b;
return sum;
```

[4 | 3 | 2 | 6]

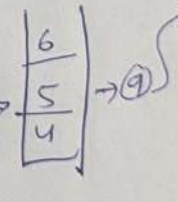
min heap →



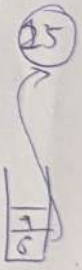
b →

a →

(5)



→ (9)



[5 + 9 + 15]

→ (29)

fractional knapsack:

given weights & values of n items.

N
W → [ ]
Val → [ ] } put them in knapsack (cap(w)) to get total max weight:

Val ⇒ [60, 100, 120]
W ⇒ [10, 20, 30]

W = 50
N = 3

1st 2nd 3rd

[50kg] bharna hai

(0) 10kg → value = ₹60
(1) 20kg → value = ₹100
(2) 30kg → value = ₹120

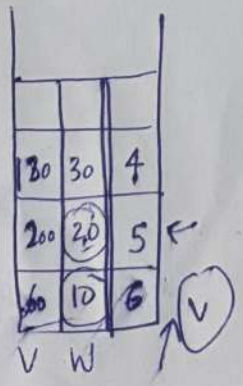
(1kg) 60/10 ⇒ (6) 100/20 ⇒ (5), 120/30 ⇒ (4) Mehnga chaia

to, → ~~60~~
[10 + 20].
60 + 100 ⇒ (160)

20/30 120/30 ⇒ (4) × 20 ⇒ (80).

160 + 80 ⇒ 240.

W = 10 + 20
⇒ W = 50 - 30 ⇒ (20) kg.



50 - 10 = 10

ans. ans + 6 × W ⇒ 60

cap 50 - 10 = (40).

(40)

5 × 20

```
Static bool cmp(pair<double, pair<int, int>> &a,
                pair<double, pair<int, int>> &b)
{
    return a.first > b.first;
}

main ---------------------------------------

vector<pair<double, pair<int, int>>> v;
for(int i=0; i<n; i++)
{
    double ratio = (double) val[i]/wt[i];
    v.push_back({ratio, {val[i], wt[i]}});
}

sort(v.begin(), v.end(), cmp)
double answer = 0.0;

for(int i=0; i<n && capacity>0; i++)
{
    int val = v[i].second.first;
    int wt = v[i].second.second;

    if(wt <= cap)
    {
        answer = answer + val;
        capacity -= weight;
    }
    else {
        answer += v[i].first * capacity;
        capacity = 0;
    }
}
return answer
}

---------------------------------------
```

Assign Cookies

→ N ⇒ child
→ every n^{th} child has greed factor
→ m cookies (4, 2, 1, 2, 1, 3) ; different sizes

greed → [1, 5, 3, 3, 4]

→ if greed [5] // cookie size must be either greater or equal to (5)

Code

```
int findContent ——
{
  sort(g.begin(),g.end());
  sort(s.begin(), s.end());
  int content=0;
  int i=0;
  int j=0;
  while(i< g.size && j< s.size())
  {
    if(g[i] <=s[i]) content++; i++,j++;
    else {
      j++;
    }
  }
  return content;
```

Lemonade Change :

→ agr 10 pay ho to 5 back krna hai
→ agr 20 pay ho to aaek (≥ 10 + 5) ka note.
→ hume cash/note ke track karke dena hai True/False.

[5, 5, 5, 10, 20]

```
int five = 0;
int ten = 0;
for(int x : bills)
{
  if(x == 5)
    five++;
  else if (x == 10)
  {
    if(five == 0) return false;
    else { ten++;
           five--;
    }
  }
  else {
    if(ten > 0 && five > 0)   // 20 hoga
    {
      ten--;
      five--;
    }
    else if (five >= 3)
    {
      five = 0;
    }
    else {
      return false;
    }
  }
return true;
}
```

Bills: | 5 | 5 | 5 | 10 | 20 |

5 = 3 × ①

10 = 1 × 0

true.