Experiment 8

Student Name: Sikander Singh Nanglu **UID: 22BET10031**
Branch: BE-IT **Section/Group: 22BET-IOT-701/A**
Semester: 6 **Date of Performance: 28/3/2025**
Subject Name:AP2 **Subject Code: 22ITP-351**

1.Aim: Implement the following problem:- Max Units on a Truck, Min Operations to make array increasing, Remove stones to Maximize total, Max Score from removing substrings, Min operations to make a subsequence, Max number of tasks you can assign.

2.Objective: To optimization problems using greedy, dynamic programming, or sorting techniques to maximize or minimize a given value under constraints.

3.Implementation/Code:

Max Units on a Truck

```
class Solution {
public:
int maximumUnits(vector<vector<int>>& boxTypes, int truckSize) {
sort(boxTypes.begin(), boxTypes.end(), [](vector<int>& a, vector<int>& b) {
return a[1] > b[1];
});
int totalUnits = 0;
for (auto& box : boxTypes) {
int count = min(box[0], truckSize);
totalUnits += count * box[1];
truckSize -= count;
if (truckSize == 0) break;
}
return totalUnits;
}
};
```

Minimum Operations to Make the Array Increasing

```
class Solution {
public:
int minOperations(vector<int>& nums) {
int operations = 0;
for (int i = 1; i < nums.size(); i++) {
if (nums[i] <= nums[i - 1]) {
operations += (nums[i - 1] - nums[i] + 1);
nums[i] = nums[i - 1] + 1;
}
}
return operations;
}
};
```

Remove Stones to Minimize the Total

```
class Solution {
public:
int minStoneSum(vector<int>& piles, int k) {
priority_queue<int> maxHeap(piles.begin(), piles.end());
while (k--) {
int top = maxHeap.top();
maxHeap.pop();
maxHeap.push(top - top / 2);
}
int total = 0;
while (!maxHeap.empty()) {
total += maxHeap.top();
maxHeap.pop();
}
return total;
}
};
```

Maximum Score From Removing Substrings

```
class Solution {
public:
int maximumGain(string s, int x, int y) {
swap(x, y);
reverse(s.begin(), s.end());
}
int score = 0;
stack<char> st;
for (char c : s) {
if (!st.empty() && st.top() == 'a' && c == 'b') {
st.pop();
score += x;
```

```
} else {
st.push(c);
}
}
string remaining;
while (!st.empty()) {
remaining += st.top();
st.pop();
}
reverse(remaining.begin(), remaining.end());
for (char c : remaining) {
if (!st.empty() && st.top() == 'b' && c == 'a') {
st.pop();
score += y;
} else {
st.push(c);
}
}
return score;
}
};
```

Minimum Operations to Make a Subsequence

```
class Solution {
public:
int minOperations(vector<int>& target, vector<int>& arr) {
unordered_map<int, int> indexMap;
for (int i = 0; i < target.size(); i++) {
indexMap[target[i]] = i;
}
vector<int> lis;
for (int num : arr) {
if (indexMap.count(num)) {
int idx = indexMap[num];
auto it = lower_bound(lis.begin(), lis.end(), idx);
if (it == lis.end()) {
lis.push_back(idx);
} else {
*it = idx;
}
}
}
return target.size() - lis.size();
}
};
```

Maximum Number of Tasks You Can Assign

```
class Solution {
public:
bool check(vector<int>& tasks, vector<int>& workers, int pills, int strength,int index)
{
multiset<int> st;
for(auto it:workers)
{
st.insert(it);
}
for(int i=index-1;i>=0;i--)
{
auto it=st.lower_bound(tasks[i]);
if(it!=st.end())
{
st.erase(it);
}
else
{
if(pills<=0)
{
return false;
}
else
{
it=st.lower_bound(tasks[i]-strength);
if(it!=st.end())
{
st.erase(it);
pills--;
}
else
{
 return false;
}
}
}
}
return true;
}
int maxTaskAssign(vector<int>& tasks, vector<int>& workers, int pills, int strength) {
sort(tasks.begin(),tasks.end());
sort(workers.begin(),workers.end());
int low=0;
int high=min(workers.size(),tasks.size());
while(low<high)
{
int mid=(low+high+1)/2;
```

```
if(check(tasks,workers,pills,strength,mid)==true)
{
low=mid;
}
else
{
high=mid-1;
}
}
return high;
}
};
```

4.Output:

Max Units on a Truck

Minimum Operations to Make the Array Increasing

Remove Stones to Minimize the Total

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

piles =
[5,4,9]

k =
2

Output

12

Expected

12

Maximum Score From Removing Substrings

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

s =
"aabbaaxybbaabb"

x =
5

y =
4

Output

20

Expected

20

Minimum Operations to Make a Subsequence

Saved

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

target =
[6,4,8,1,3,2]

arr =
[4,7,6,2,3,8,6,1]

Output

3

Expected

3

Maximum Number of Tasks You Can Assign

5.Learning Outcomes:-

- Ability to analyze problems, evaluate information, and make logical decisions.
- Capability to identify, understand, and develop solutions to complex issues.
- Proficiency in expressing ideas clearly, both verbally and in writing
- Willingness to learn new skills and adjust to changing environments.
- Ability to work effectively with others in diverse environments.