



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

EXPERIMENT - 8

Student Name: Yash Dixit

Branch: BE-CSE

Semester: 6th

Subject Name: Advance Programming

UID: 22BCS10730

Section/Group: IOT-605'A'

Date of Performance: 02/04/25

Subject Code: 22CSP-367

1. Maximum Units on a Truck (Amazon)

[Link](#)

python

CopyEdit

```
def maximumUnits(boxTypes, truckSize):
    boxTypes.sort(key=lambda x: -x[1])
    units = 0
    for box, unit in boxTypes:
        if truckSize >= box:
            units += box * unit
            truckSize -= box
        else:
            units += truckSize * unit
            break
    return units
```

2. Minimum Operations to Make the Array Increasing (Amazon)

[Link](#)

python

CopyEdit

```
def minOperations(nums):
    operations = 0
    for i in range(1, len(nums)):
        if nums[i] <= nums[i-1]:
            diff = nums[i-1] - nums[i] + 1
            nums[i] += diff
            operations += diff
    return operations
```

3. Remove Stones to Maximize Total (Nvidia)

[Link](#)

python

CopyEdit

import heapq

```
def maxStoneSum(piles, k):
    piles = [-pile for pile in piles]
    heapq.heapify(piles)
    for _ in range(k):
        largest = -heapq.heappop(piles)
        largest -= largest // 2
        heapq.heappush(piles, -largest)
    return -sum(piles)
```



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Day 16 (Greedy - Hard)

4. Maximum Score from Removing Substrings (Adobe)

[Link](#)

python

CopyEdit

```
def maximumGain(s, x, y):
    res = 0
    if x > y:
        first, second = "ab", "ba"
        fx, fy = x, y
    else:
        first, second = "ba", "ab"
        fx, fy = y, x

    def remove(s, a, score):
        stack = []
        total = 0
        for c in s:
            if stack and stack[-1] == a[0] and c == a[1]:
                stack.pop()
                total += score
            else:
                stack.append(c)
        return "".join(stack), total

    s, gain = remove(s, first, fx)
    res += gain
    s, gain = remove(s, second, fy)
    res += gain

    return res
```

5. Minimum Operations to Make a Subsequence (Adobe)

[Link](#)

python

CopyEdit

import bisect

```
def minOperations(target, arr):
    mapping = {val: idx for idx, val in enumerate(target)}
    temp = [mapping[x] for x in arr if x in mapping]

    lis = []
    for num in temp:
        idx = bisect.bisect_left(lis, num)
        if idx == len(lis):
            lis.append(num)
        else:
            lis[idx] = num
    return len(target) - len(lis)
```

6. Maximum Number of Tasks You Can Assign (Nvidia)

[Link](#)

python



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

CopyEdit

import bisect

```
def maxTaskAssign(tasks, workers, pills, strength):
```

```
    tasks.sort()
```

```
    workers.sort()
```

```
    def can_assign(k):
```

```
        t = tasks[:k]
```

```
        w = workers[-k:]
```

```
        i = 0
```

```
        j = k-1
```

```
        p = pills
```

```
        while i < k and j >= 0:
```

```
            if w[j] >= t[i]:
```

```
                j -= 1
```

```
                i += 1
```

```
            elif p > 0 and w[j] + strength >= t[i]:
```

```
                p -= 1
```

```
                j -= 1
```

```
                i += 1
```

```
            else:
```

```
                i += 1
```

```
        return j == -1
```

```
    low, high = 0, min(len(tasks), len(workers))
```

```
    res = 0
```

```
    while low <= high:
```

```
        mid = (low + high) // 2
```

```
        if can_assign(mid):
```

```
            res = mid
```

```
            low = mid + 1
```

```
        else:
```

```
            high = mid - 1
```

```
    return res
```