

1) Randomized, Dynamic Programming, Greedy, Ad Hoc, Divide and Conquer, Brute Force.

- a. Greedy
- b. Greedy
- c. Randomize (Greedy)
- d. Divide and Conquer
- e. Brute Force (Back Tracking sort of)

2) We count the number of 3 pages groups.

If # 2 pages is more than # 1 pages we should always punch a 1 pages group with a 2 pages group to prevent the extra 1 pages groups at the end.

If we have more 2 page groups than 1 page group.

we can punch the 1 pages with the 2 pages.

Answer become # 2 page groups + # 3 page groups

If we have less than to minimize the number of 1 page remaining after each 2 page is punched we punch a 1 page with a 2 page when possible.

Otherwise we punch as many 1 page groups with 2 pages as possible. # 3 pages + # 2 pages + Ceiling ((#1 pages - # 2pages) / 3)

3)

Check if we have n gifts of the same value return true if we do.

Sort the values. Then distribute the lowest gift with the highest gift. Repeat this process for the next lowest/next highest. If we have a different value for any sum we return false otherwise return true.

Proof Assume there is a way to distribute gifts evenly without doing the above method. Find the first differing pair and the sum of the gifts of that pair will be less than the sum of the gifts of the last differing pair since the sum of the sorted values is increasing.

4) The five characters are "a" "b" "d" "n" "_"

A = 16, B = 8, D = 4, N = 7, _ = 11

See Written notes?

5)

Seen values: Y, V, W, T, Q, R, U, S, X, Z

On Stack: Y, V, Z

Order: Y, V, W, T, Q, R, U, S, X, Z

6) Forward, Cross, Back, Tree

7)

```
topo_elements[position++] = cur;
for () {
    inDeg[e.en]--;
    if ((inDeg[e.en] == 0) && !visited[e.en]) {
        visited[e.en] = true;
        DFSTopo(e.en , g, visited, inDeg);
    }
}
```

8) See Written notes

9) 5!

10) See Written notes

11) Mult part

- a. Not done
- b. Not done
- c. See Written notes