

PAS Problem 1

Input: city, country
⋮

$O(n \log n)$

APP #1,
→ sort by country name

→ gather by country,
& output (string sorting)

Output:

Country
cities, ...
country.
⋮

$O(n)$

#2
→ hash by country name
→ create/maintain a linked list for each hash value
→ linked list of cities

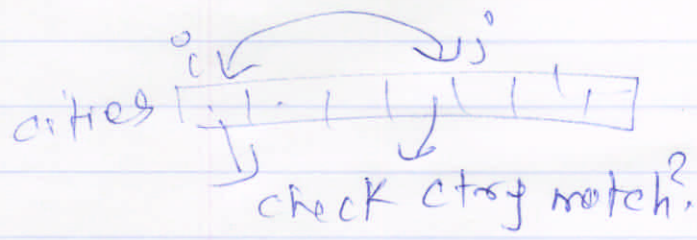
→ STL hash-map
<key, value>

country name → linked list

#3

Union-find

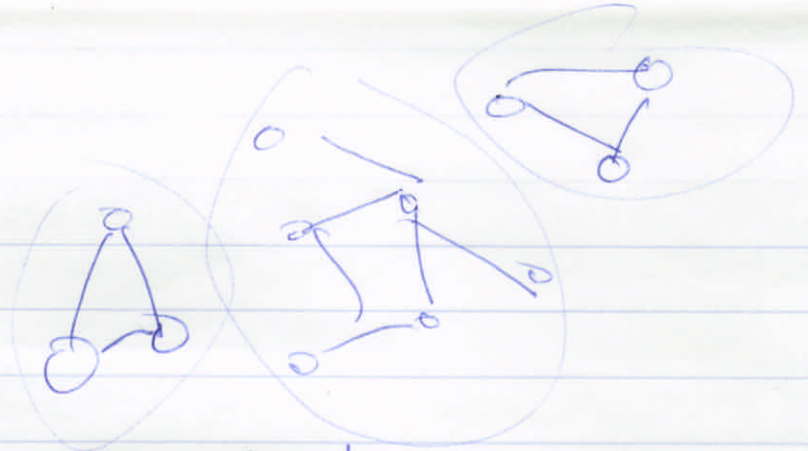
for (i = 1 to n)
for (j = i+1 to n);
if (match)
union(i, j) $O(n^2)$



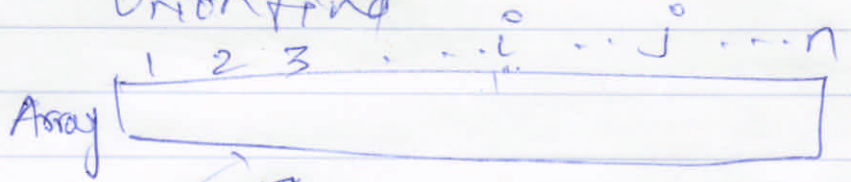
Prob 2

city₁, city₂

<string, string>



Union find



```

1) hash
for # city name
    hash["city name"]
    ↓
    insert
id = 1;
if hash.insert("city1")
    succeeds
    => id++;
else
    ignore;

```

```

2) init UF →
    ecn) (call constructor)
3) For (i = 0; i < n; i++)
    for (j = i+1; j < n; j++)
        if (i conn j)
            Union(i, j)

```

③

Linearly scan input file ↓

$O(\#edges \times \log n)$ + line (i, j) → Union(i, j)

$O(\#lines \times \#edges)$

HW 6

1) (A)

For (i=1, i ≤ n; i++)

For (j=i+1, j ≤ n; j++)

If (year(i) == year(j))

Union(i, j)

$O(n^2)$

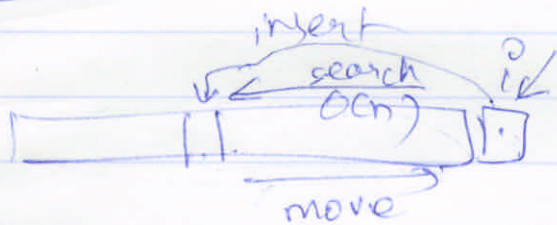
(B)

Use integer sort
(counting)
sort.

$O(n)$



2)



worst case

→ old insert $O(n^2)$

→ New search $O(\lg n)$

worst case move $O(n)$

worst case $O(n^2)$

→ practice → some improvements
little.

3)



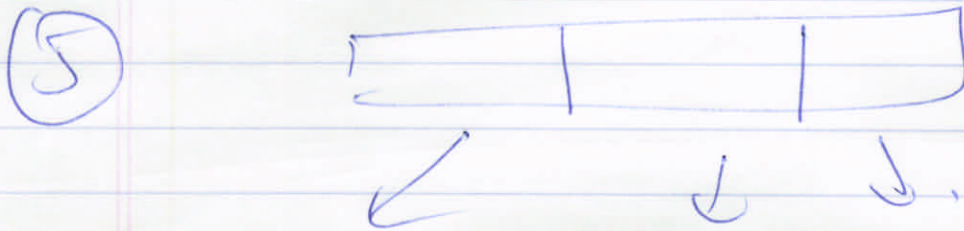
(i) Ins sort $O(n^2)$

(ii) Merge $O(n \lg n)$

(iii) Quick

↙ ↘ $O(n \lg n)$

- ④
- a) Yes
 - b) No
 - c) No
 - d) Yes
 - e) Yes.



$$\begin{aligned}
 T(n) &= 3T\left(\frac{n}{3}\right) + n \\
 &= 3\left[3T\left(\frac{n}{3^2}\right) + \frac{n}{3}\right] + n \\
 &= 3^2T\left(\frac{n}{3^2}\right) + n + n \\
 &\vdots \\
 &= 3^k T\left(\frac{n}{3^k}\right) + kn
 \end{aligned}$$

$$\begin{aligned}
 k = \log_3 n \Rightarrow T(n) &= n + n(\log_3 n) \\
 &= O(n \log n)
 \end{aligned}$$