Final Test Algorithm

by Manuel Morales

Q1:If you need the algorithm X, and in the language that you use there is an implementation for said algorithm. In which case(s) would you implement your own version of the algorithm?

Cuando el algoritmo que se necesita requiere una implementacion muy pero muy especifica y nosotros podemos mejorar la implementacion que ya existe para asi tomar ventaja de la situacion.

Q2:Explain why would we want to consider the average-case analysis rather than the worst-case analysis

Porque en la mayoria de los casos, el peor caso no se va a dar, y si se da, no va a ser tan frecuente como los otros casos, por lo que es mejor analizar el caso promedio, ya que es el que mas se va a dar.

Q3

```
/**
  * Time Complexity: O(log(min(a, b)))
  */
 public int method(int a, int b) {
   while (b != 0) {
     int tmp = b;
     b = a % b;
     a = tmp;
   }
   return a;
 }
```

Q4

```
#ifdef LOCAL
#include "/home/morven/Desktop/code/manuel-competitive/conf/debug.h"
#define line cerr << "-------------------" << endl;
#else
#define deb(x...)
#define line
#endif

#include <bits/stdc++.h>
using namespace std;

template <typename... T>
void cinn(T &...args)
```

```
{
  ((cin >> args), ...);
}

template <typename... T>
void coutt(const T &...args)
{
  __typeof(sizeof...(T)) i = 1;
  ((cout << args << (i++ != sizeof...(T) ? " " : "")), ...);
  cout << '\n';
}

template <typename T>
void couts(const T &xs)
{
  __typeof(xs.size()) i = 1;
  for (auto &x : xs)
  {
    cout << x << (i++ == xs.size() ? '\n' : ' ');
  }
}

#define endl '\n'
#define ll long long
#define F first
#define S second
#define pb push_back
#define sz size
#define all(x) begin(x), end(x)
#define sortt(x) sort(all(x))
#define each(x, xs) for (auto &x : (xs))
#define rep(i, be, en) for (__typeof(en) i = (be); i < (en); i++)
#define per(i, be) for (__typeof(be) i = (be)-1; i >= 0; i--)
#define strInt(str) stoi(str)
#define chrInt(chr) (int)chr - 48
#define ensp(i, n) (i == n - 1 ? '\n' : ' ')
#define readline(x) getline(cin, x)

const long long int INF = INT64_MAX;
const long long int MOD = 1e9 + 7;
const long double PI = acos(-1);
const vector<int> DX{1, 0, -1, 0}, DY{0, 1, 0, -1};

void _()
{
  ios::sync_with_stdio(0);
  cin.tie(0);
  cout.tie(0);
#ifdef LOCAL
  freopen("/home/morven/Desktop/code/manuel-competitive/conf/main.in", "r",
stdin);
  freopen("/home/morven/Desktop/code/manuel-competitive/conf/main.out",
"w", stdout);
  freopen("/home/morven/Desktop/code/manuel-competitive/conf/main.err",
```

```
"w", stderr);
#endif
}

void solve();

int main()
{
  _();
  int T;
  T = 1;
  // cin >> T;
  rep(t, 0, T) { solve(); }
  return 0;
}

const int MAXN = 202020;
int n, m, y, x;
string str;
unordered_map<string, pair<int, priority_queue<int, vector<int>,
greater<int>>>> mp;

/**
 * Time Complexity: O(n log n)
 *
 * Elegi un unordered map, para tener mayor rapidez al momento de buscar
las
 * llaves y los valores, ya que el tiempo de busqueda es O(1) en promedio.
 *
 * Elegi una priority queue de menor a mayor, para que al momento de hacer
pop
 * me devuelva el menor valor, y asi poder hacer la comparacion con el
valor
 * que se quiere insertar, de esa manera si el valor que se quiere insertar
 * es mayor al que esta en el tope, se hace pop y se inserta el nuevo
valor.
 * y si el valor que se quiere insertar es menor al que esta en el tope, no
 * se inserta.
 * Y de esta forma tambien puedo mantener la suma total de los valores que
 * estan en la priority queue de manera constante, ya que al hacer pop, se
 * resta el valor que se saco, y al hacer push se suma el valor que se
inserto.
 *
 * Space Complexity: O(n)
 *
 * Otra solucion seria usar un multiset, ya que tambien mantiene los
valores
 * ordenados, y al momento de insertar un nuevo valor, se inserta en O(log
n),
 * y al momento de hacer pop, se hace en O(log n), y al momento de hacer la
 * comparacion, se hace en O(1), ya que el valor que se quiere insertar es
 * mayor al que esta en el tope, se hace pop y se inserta el nuevo valor.
 * y si el valor que se quiere insertar es menor al que esta en el tope, no
 * se inserta.
```

```
 */
void solve()
{
  cinn(n, m);
  rep(i, 0, m)
  {
    cinn(str, x);
    if (mp[str].S.size() == n)
    {
      if (mp[str].S.top() >= x)
      {
        continue;
      }
      mp[str].F -= mp[str].S.top();
      mp[str].F += x;
      mp[str].S.pop();
      mp[str].S.push(x);
      continue;
    }
    mp[str].F += x;
    mp[str].S.push(x);
  }
  cinn(y);
  rep(i, 0, y)
  {
    cinn(str, x);
    if (mp[str].S.size() == n)
    {
      if (mp[str].S.top() >= x)
      {
        continue;
      }
      mp[str].F -= mp[str].S.top();
      mp[str].F += x;
      mp[str].S.pop();
      mp[str].S.push(x);
      continue;
    }
    mp[str].F += x;
    mp[str].S.push(x);
  }

  priority_queue<pair<int, string>> pq;
  each(e, mp)
  {
    pq.push({e.S.F, e.F});
  }

  int count = 0;
  while (!pq.empty() && count < 3)
  {
    coutt(pq.top().S);
    pq.pop();
    count++;
```

```
    }
  }
```