UNIVERSIDAD TORCUATO DI TELLA

5/5/2023

Algoritmos y Estructuras de Datos – Primer Parcial

Licenciatura en Tecnología Digital, UTDT

Primer semestre 2023

- El examen es individual.
- No está permitido comunicarse por ningún medio con otros estudiantes ni con otras personas durante el examen, excepto con los docentes de la materia.
- Puede consultarse a los docentes solo por aclaraciones específicas del enunciado.
- Cada ejercicio debe resolverse en hoja aparte.

Problema 1. (30 puntos)

Dado un `vector<string>` v y un `string` s **que debe aparecer como subcadena de algún** $v[i]$, se desea especificar el problema `buscar`, que devuelve un `vector<int>`.

- En la posición i del `vector<int>` de salida se debe indicar la posición donde aparece s en el string $v[i]$ de entrada; si s no aparece se debe indicar `|v[i]|`.

Por ejemplo, para $v =$ {"hola", "volar", "olas", "olaolaola", "amigo"} y $s =$ "ola" se debe devolver $res =$ {1,1,0,0,5}.

(a) Especificar el problema `buscar`.

Problema 2. (20 puntos)

La función `chequear` *toma un vector de strings* `v` *y devuelve* **bool**. *Debe devolver* `true` *si y sólo si la cantidad de strings en* `v` *que empiezan con la letra* `"a"` *es par. Por ejemplo, para* `v = {"sad","ars","pip","a*","amlo"}` *se debe devolver* `false` *porque la cantidad de strings que comienzan con* `"a"` *es 3.*

```
bool chequear(const vector<string> & v)
```

Pre: $(\forall j : \texttt{int})(0 \leq j < |v| \implies |v[j]| > 0)$

Post: $res = \texttt{true} \iff \left(\sum_{j=0}^{|v|-1} \beta(v[j][0] = \texttt{"a"}) \mod 2 = 0\right)$

```
1 int i = v.size();
2 int count = 0;
3 while (0 < i){
4   i = i - 1;
5   if (v[i][0] == "a")
6     count = count + 1;
7 }
8 bool res = count % 2 == 0;
```

(a) Dar una precondición P_c y una postcondición Q_c adecuadas para el ciclo.

(b) Dar un invariante del ciclo $\mathcal{I}$ que permita demostrar la corrección del programa. Explicar en palabras por qué el invariante es adecuado.

(c) Dar un ejemplo de valores de `count`, `v`, y `i` que cumpla con $\mathcal{I}$ y un ejemplo que no lo cumpla.

(d) Demostrar que $\mathcal{I} \wedge B = \texttt{false} \implies Q_c$, donde B es la guarda del ciclo.

Problema 3. (20 puntos)
Dado un vector de enteros, se desea dibujar su *histograma en el rango [a, b]*. El *histograma* de un conjunto de valores es un gráfico de barras verticales donde cada barra tiene como longitud la cantidad de apariciones de cada valor del rango [a, b]. Nosotros representaremos el histograma como un `vector<int>` con la longitud de cada barra en el intervalo [a,b], que incluye a y b. Notar que el vector de salida res siempre tendrá longitud $b-a+1$, y que, por ejemplo, $res[0]$ y $res[b-a]$ almacenan la cantidad de apariciones de a y la cantidad de apariciones de b respectivamente.

Ejemplo: el histograma en el rango [2, 5] del vector `{1,1,1,5,5,2,2,3,3,5,5,10}` se representará con el `vector<int>` `{2,2,0,4}`. Para el mismo vector de entrada, su histograma en el intervalo [-2,4] será `{0,0,0,3,2,2,0}`.

(a) Escribir una función con un único ciclo que, dado un `vector<int>` v, y dos enteros a, b con $a < b$, devuelva un `vector<int>` que represente su histograma entre a y b. El vector v sólo debe recorrerse una vez.

Sugerencia: Se puede crear un vector de N ceros con la expresión `vector<int>(N,0)`.

(b) Dar la especificación completa del ciclo propuesto ($\mathcal{P}_c, \mathcal{Q}_c, \mathcal{I}, fv$). Justificar por qué es adecuada para el código escrito.

Problema 4. (30 puntos) Se quiere implementar la función

```
string concatenar_palindromos(const vector<string> & v)
```

que debe concatenar en un único `string` de salida todos los elementos de `v` que sean palíndromos. Por ejemplo, si `v = {"perro", "amama", "rima", "somos", "anilina"}`, la función debe devolver `"amamasomosanilina"`. Afortunadamente, se cuenta con la función `bool es_palindromo(string s)` ya implementada, que debe utilizarse como función auxiliar.

(a) Implementar `concatenar_palindromos` con recursión simple, es decir utilizando un sólo llamado recursivo.

(b) Implementar `concatenar_palindromos` con la técnica Divide & Conquer, partiendo el problema en 2 y utilizando 2 llamados recursivos.

En cada caso, si necesita cambiar los parámetros de la función debe hacerlo en una función auxiliar y **mostrar cómo le llama a esa función auxiliar desde la función `concatenar_palindromos` original.** Recuerde que en `C++` se puede utilizar el operador `+` para concatenar dos `strings`.