

CS2013 - Programación III  
Práctica Calificada #3 (PC3)  
2023 - 1

Profesor: Rubén Rivas

Alumno:

---

Indicaciones:

- La Duración es de 100 minutos.
  - La evaluación consta de 3 preguntas.
  - Se permite el uso de calculadora científica y tablas, copias, apuntes, libros y toda información necesaria.
- 

**Hash - 7 puntos**

Diseñar y desarrollar el template de función **can\_text\_split** que confirme si es posible dividir un texto en palabras contenidas en una lista (contenedor generico).

**Caso de uso #1**

```
vector<string> vec = {"este", "texto", "UTEC", "prueba"};
string text1 = "estetexto";
cout << boolalpha << can_text_split(text1, vec) << endl;
```

**Caso de uso #2**

```
list<string> lst = {"este", "texto", "UTEC", "prueba"};
string text2 = "estetextoUTEC";
cout << boolalpha << can_text_split(text2, lst) << endl;
```

**Caso de uso #3**

```
cout << boolalpha
    << can_text_split<string>("prueba",
        {"este", "texto", "UTEC", "prueba"}) << endl;
```

#### Caso de uso #4

```
cout << boolalpha  
    << can_text_split<wstring>(L"estaprueba",  
        {L"esta", L"texto", L"UTEC", L"prueba"}) << endl;
```

#### Tree – 7 puntos

Diseña un método que permita calcular la maxima profundidad de un arbol.

#### Caso de uso #1

```
binary_search_tree<string, int> bst;  
bst.insert({"B", 20});  
bst.insert({"A", 10});  
cout << bst.max_depth() << endl;    // 2
```

#### Caso de uso #2

```
binary_search_tree<string, int> bst;  
bst.insert({"B", 20});  
bst.insert({"A", 10});  
bst.insert({"D", 30});  
bst.insert({"C", 40});  
bst.insert({"E", 35});  
cout << bst.max_depth() << endl;    // 3
```

#### Caso de uso #3

```
binary_search_tree<string, int> bst;  
bst.insert({"A", 20});  
bst.insert({"B", 10});  
bst.insert({"C", 30});  
bst.insert({"D", 40});  
bst.insert({"E", 35});  
cout << bst.max_depth() << endl;    // 5
```

### Algoritmos de Grafos - 6 puntos

Utilizando el algoritmo de **Prim** e iniciando en el vértice **G** determinar el primer árbol expandido mínimo.

Adicionalmente realizar el recorrido a lo largo (BFS) y el recorrido a profundidad (DFS) utilizando los algoritmos correspondientes.

Sustentar claramente sus respuestas con el procedimiento de cálculo.



