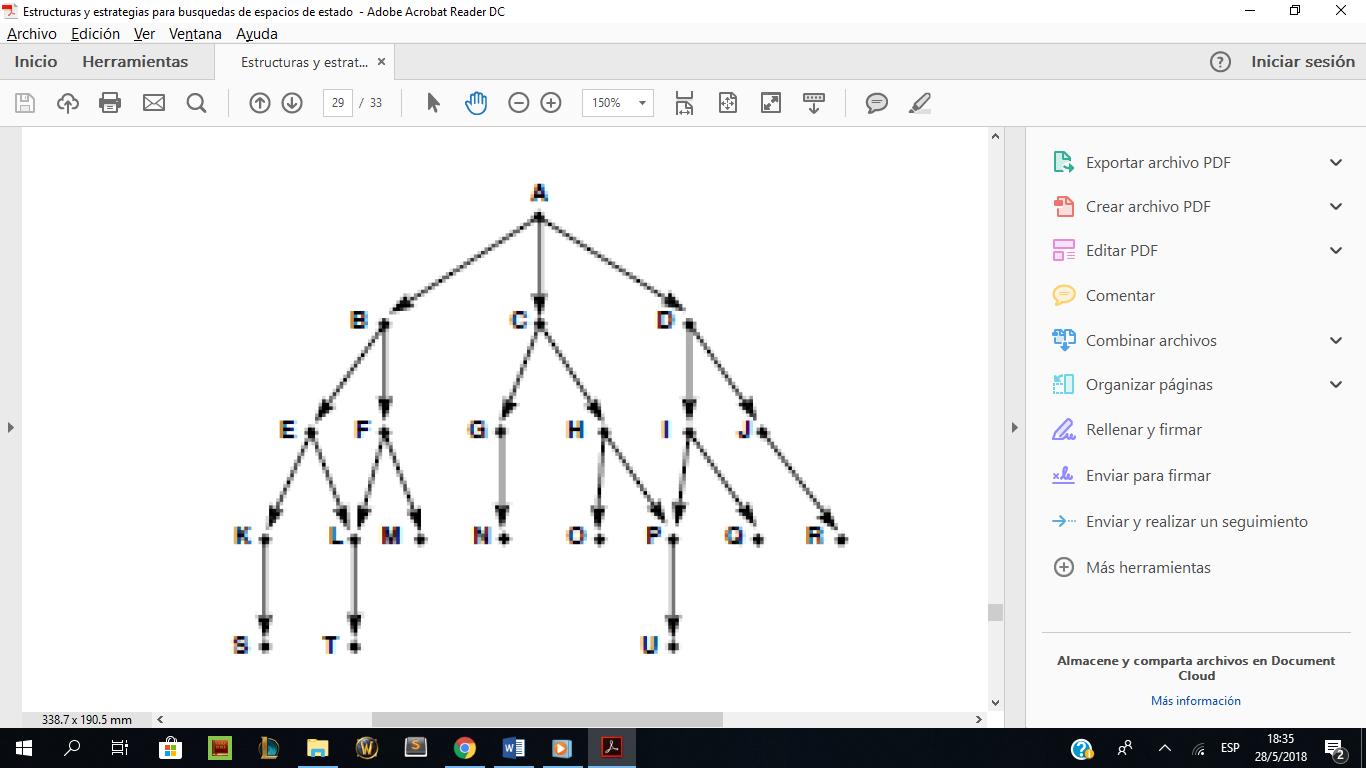
Deber 4



**Parte A:**

Usando el algoritmo Breadth first search, realice todas las pruebas de escritorio necesarias numerándolas. Por cada pasada del algoritmo hasta que identifique la ruta objetivo, identifique por cada corrida (pasada por el algoritmo)

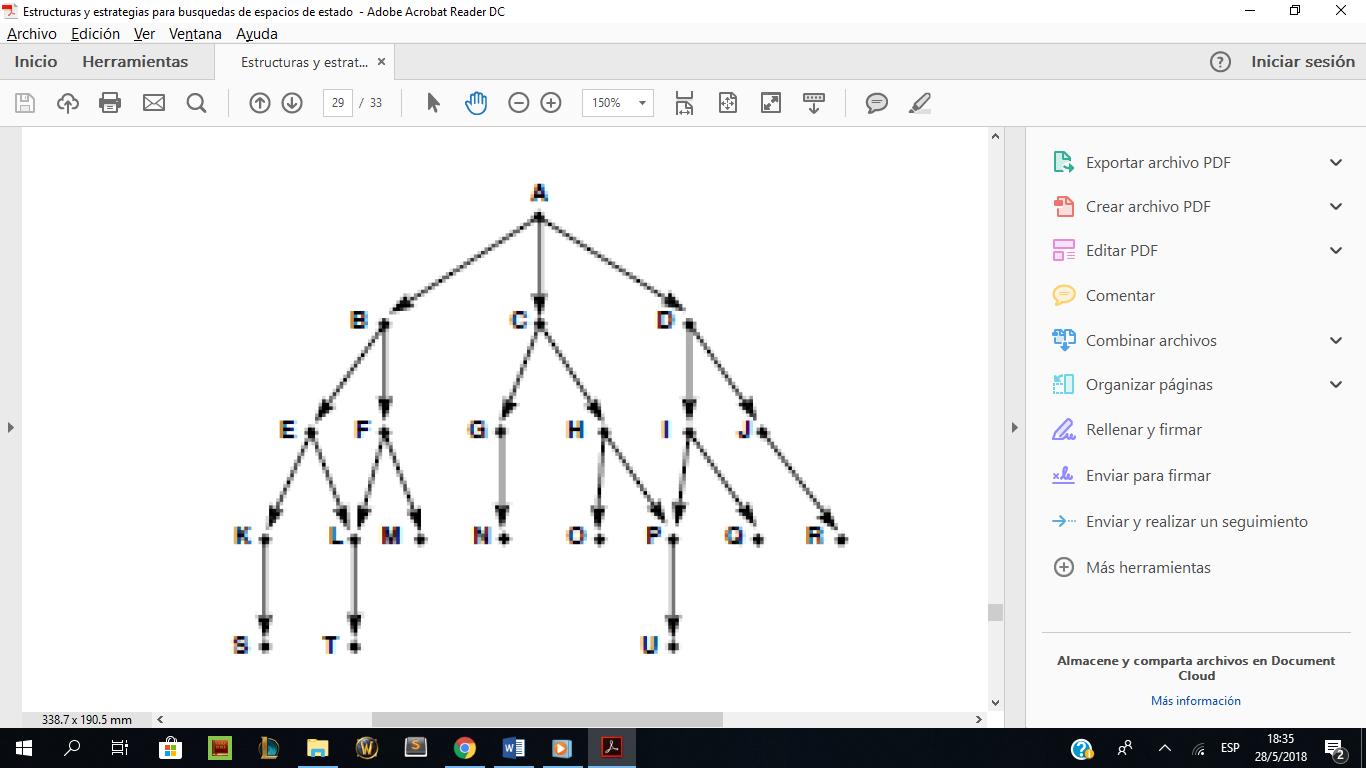
* El estado de la lista Open
* El valor de la X
* Los hijos de X
* El estado de la lista Closed

|  |  |  |  |
| --- | --- | --- | --- |
| **Open** | **Closed** | **X** | **SX** |
| A |  |  |  |
| B, C, D | A | A | B, C, D |
| C, D, E, F | A, B | B | E, F |
| D, E, F, G, H | A, B, C | C | G, H |
| E, F, G, H, I, J | A, B, C, D | D | I, J |
| F, G, H, I, J, K, L | A, B, C, D, E | E | K, L |
| G, H, I, J, K, L, M | A, B, C, D, E, F | F | M |
| H, I, J, K, L, M, N | A, B, C, D, E, F, G | G | N |
| I, J, K, L, M, N, O, P | A, B, C, D, E, F, G, H | H | O, P |
| J, K, L, M, N, O, P, Q | A, B, C, D, E, F, G, H, I | I | Q |
| K, L, M, N, O, P, Q, R | A, B, C, D, E, F, G, H, I, J | J | R |
| L, M, N, O, P, Q, R, S | A, B, C, D, E, F, G, H, I, J, K | K | S |
| M, N, O, P, Q, R, S, T | A, B, C, D, E, F, G, H, I, J, K, L | L | T |
| N, O, P, Q, R, S, T | A, B, C, D, E, F, G, H, I, J, K, L, M | M |  |
| O, P, Q, R, S, T | A, B, C, D, E, F, G, H, I, J, K, M, N | N |  |
| P, Q, R, S, T | A, B, C, D, E, F, G, H, I, J, K, M, N, O | O |  |
| Q, R, S, T, U | A, B, C, D, E, F, G, H, I, J, K, M, N, O, P | P | U |
| R, S, T, U | A, B, C, D, E, F, G, H, I, J, K, M, N, O, P, Q | Q |  |
| S, T, U | A, B, C, D, E, F, G, H, I, J, K, M, N, O, P, Q, R | R |  |
| T, U | A, B, C, D, E, F, G, H, I, J, K, M, N, O, P, Q, R, S | S |  |
| U | A, B, C, D, E, F, G, H, I, J, K, M, N, O, P, Q, R, S, T | T |  |
|  | A, B, C, D, E, F, G, H, I, J, K, M, N, O, P, Q, R, S, T | U |  |

**Parte B:**

Usando la función Depth first search, realice todas las pruebas de escritorio necesarias hasta identificar la ruta objetivo. Por cada pasada de la función hasta que identifique la ruta objetivo, identifique en cada corrida

* El estado de la lista Open
* El valor de la X
* El hijo de X
* El estado de la lista closed



|  |  |  |  |
| --- | --- | --- | --- |
| **Open** | **Closed** | **X** | **SX** |
| A |  |  |  |
| D, C, B | A | A | B, C, D |
| J, I, C, B | A, B, D | D | I, J |
| R, I, C, B | A, B, D, J | J | R |
| I, C, B | A, B, D, J, R | R |  |
| Q, P, R, C, B | A, B, D, J, R, I | I | P, Q |
| P, R, C, B | A, B, D, J, R, I, Q | Q |  |
| U, R, C, B | A, B, D, J, R, I, Q, P | P | U |
| R, C, B | A, B, D, J, R, I, Q, P | U |  |
| C, B | A, B, D, J, R, I, Q, P, R | R |  |
| H, G, B | A, B, D, J, R, I, Q, P, R, C | C | G, H |
| O, G, B | A, B, D, J, R, I, Q, P, R, C, H | H | O |
| G, B | A, B, D, J, R, I, Q, P, R, C, H, O | O |  |
| N, B | A, B, D, J, R, I, Q, P, R, C, H, O, G | G | N |
| B | A, B, D, J, R, I, Q, P, R, C, H, O, G | N |  |
| F, E | A, B, D, J, R, I, Q, P, R, C, H, O, G, B | B | E, F |
| M, L, E | A, B, D, J, R, I, Q, P, R, C, H, O, G, B, F | F | L, M |
| L, E | A, B, D, J, R, I, Q, P, R, C, H, O, G, B, F, M | M |  |
| T, E | A, B, D, J, R, I, Q, P, R, C, H, O, G, B, F, M, L | L | T |
| E | A, B, D, J, R, I, Q, P, R, C, H, O, G, B, F, M, L, T | T |  |
| K | A, B, D, J, R, I, Q, P, R, C, H, O, G, B, F, M, L, T, E | E | K |
| S | A, B, D, J, R, I, Q, P, R, C, H, O, G, B, F, M, L, T, E, K | K | S |
|  | A, B, D, J, R, I, Q, P, R, C, H, O, G, B, F, M, L, T, E, K, S | S |  |

**Parte C:**

Compare los dos resultados:

Depth first search: 23 pasos.

Breadth first search: 21 pasos.

En cuestión de tiempo, Breath first search es mejor.

Tamaño máximo de la lista open en Depth first search: 5

Tamaño máximo de la lista open en Breadth first search: 8

En cuestión de memoria de la lista open, Breadth first search es mejor.

Tamaño máximo de la lista closed en Depth first search: 21

Tamaño máximo de la lista closed en Breadth first search: 19

En cuestión de memoria de la lista closed, Depth first search es mejor.

Cuál de las dos funciones es más eficiente:

Dado los casos comparados de memoria y tiempo para este grafo, el algoritmo breadth first search es mejor.

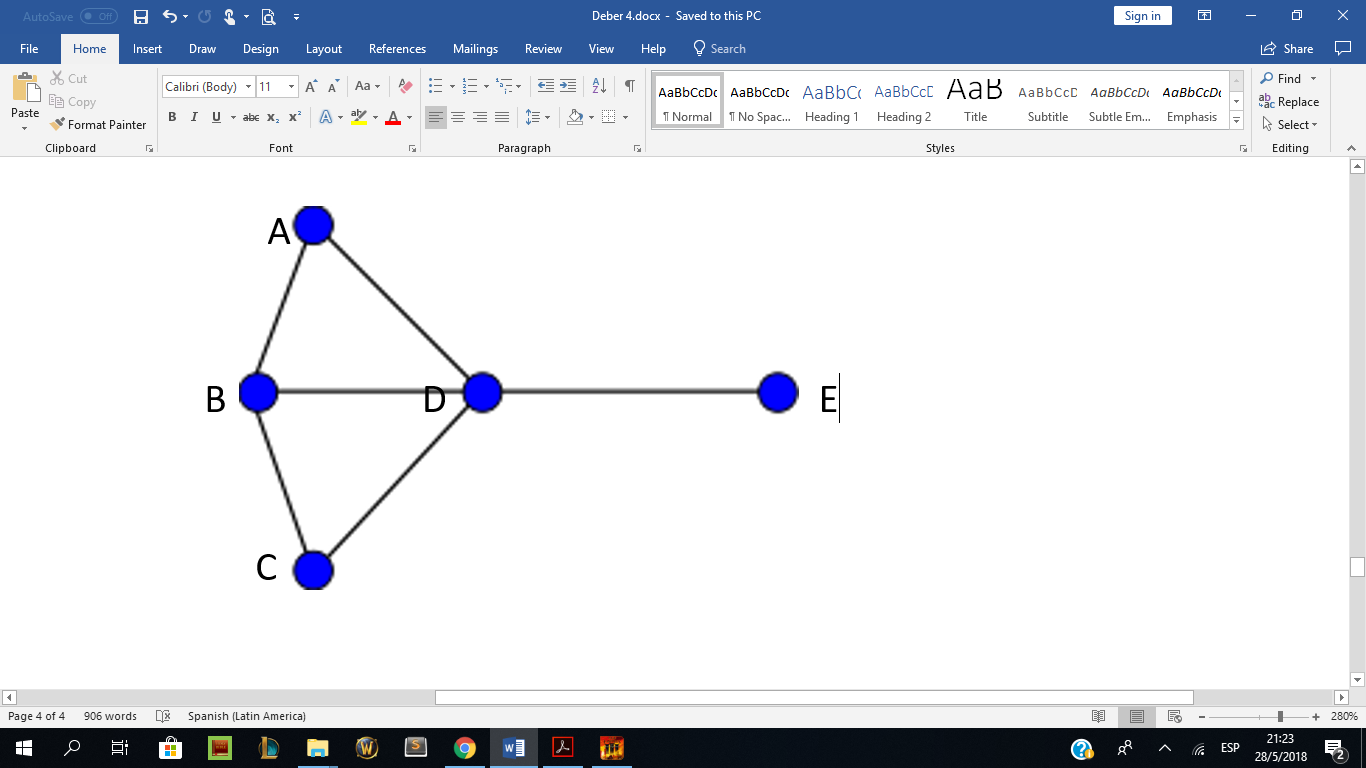
De qué depende que las funciones sean más eficientes. ¿Cuáles deberían ser las características del grafo?

De la posición en la que se encuentre el objetivo (goal) en el grafo, del tiempo de búsqueda y del espacio de memoria.

Que todos sus nodos estén conectados, y siempre haya un camino que conecte cualquier par de nodos en el grafo.

Que la respuesta que estoy buscando esté en el grafo, de no ser así, puedo quedarme atrapado en un lazo infinito.

Dibuje un grafo nuevo, con una ruta objetivo que puede ser encontrada rápidamente, para la función de deep first search o breath first search.



|  |  |  |  |
| --- | --- | --- | --- |
| **Open** | **Closed** | **X** | **SX** |
| A |  |  |  |
| B, D | A | A | B, D |
| D, C | A, B | B | C |
| C, E | A, B, D | D | E |
| E | A, B, D, C | C |  |
|  | A, B, D, C | E |  |