

1. Variable and Constant Declaration Defects (VC)

1. Si, por ejemplo línea 11.
2. No, de las líneas 11 a la 19, no hay confusión.
3. No, *dijkstraExec* no se inicializa al principio.
4. No, las variables de las líneas 11 a 19 se utilizan en varias funciones.
5. No, no hay.
6. No, no hay macros en Java.
7. No, puede variar el número de vértices.

2. Function Definition Defects (FD)

8. No, en la línea 73, la función *nextCur*, no deja claro con ese nombre el propósito de la función.
9. Si los checkea.
10. No, en la línea 63, la función *initializeDataStructures* no debe retornar nada.

3. Class Definition Defects (CD)

11. Si, línea 41. Constructor Dijkstra.
12. Tan solo hay una clase.
13. No, tan solo hay una clase.
14. No, tan solo hay una clase.

4. Computation/Numeric Defects (CN)

15. No se produce overflow porque no hay una carga de datos excesiva. No se produce underflow, ya que no hay un procesamiento mayor a la carga.
16. No, línea 118 es más probable que falle.
17. Hay comentarios que evitan la ambigüedad pero no paréntesis con ese fin.

5. Comparision/Relational Defects (CR)

18. No en su totalidad.
19. No, probablemente la variable *visitado* no sea correcta siempre.
20. Sí, línea 118.

6. Control Flow Defects (CF)

21. Si, ejemplo línea 77.

- 22. Si, linea 165.
- 23. Si, no tenemos bucles con más de una salida.
- 24. No hay ningún switch.
- 25. No hay ningún switch.
- 26. Si, es correcta pero no muy profundo.
- 27. No, no sería necesario. Línea 113
- 28. No tenemos null bodied control structures.
- 29. Si, terminan todas.
- 30. Si, se evitan porque no se encuentran en el código.

7. Input-Output Defects (IO)

- 31. No se utilizan ficheros.
- 32. No se utilizan ficheros.
- 33. No se utilizan ficheros.
- 34. No se utilizan ficheros.
- 35. No se utilizan ficheros.
- 36. No se utilizan ficheros.

8. Module Interface Defects (MI)

- 37. Si, ya que no hay llamadas entre las funciones.
- 38. Si, ya que no tenemos este tipo de valores.

9. Comment Defects (CM)

- 39. Si, todo tiene comentarios.
- 40. No, en las líneas 74 y 75.
- 41. Si, se entiende correctamente.
- 42. Si, es consistente.
- 43. Si, están de acuerdo.
- 44. Si ayudan.
- 45. Si, son suficientes.
- 46. Hay los justos y necesarios.

10. Packing Defects (LP)

- 47. No hay archivos.
- 48. Si, son menores a 60.
- 49. Si, son menores a 2000.

11. Modularity Defects (MO)

- 50. Solo hay un paquete.
- 51. Solo hay un paquete.
- 52. No.
- 53. Si, se usa ArrayList y es correcto su uso. Linea 159.

12. Performance Defects (PE) [Optional]

- 54. El algoritmo de Dijkstra es el mejor para este problema.
- 55. Si, ya que no se han gastado recursos para hacer pruebas.
- 56. Sí, sin duda.
- 57. No, no se guardan.
- 58. Se podría mover pero dejaría de ser Dijkstra.
- 59. No, todas aportan al resultado final.
- 60. Si se puede deshacer. Por ejemplo el bucle de la línea 77 en la función nextCur.
- 61. No, no hay.