

# Casos De Prueba

Nombre: Caso\_01\_no\_detecta\_la\_posicion\_del\_bus\_correctamente\_y\_la\_escuela

Descripción: busca detectar un error al momento de leer los archivos.

CAMBIO.IN	CAMBIO.OUT(Esperado)	CAMBIO.OUT(Obtenido)
8 2 7 13 2 4 4 2 1 5 2 3 2 3 1 3 4 1 4 7 5 3 1 5 3 1 6 7 8 4 2 6 8 1 8 7 6 6 7 8 5 3 2	7 13 6	7 13 6

Informe:No presenta error.

Nombre: Caso\_02\_no\_leé\_las\_calles\_y\_esquinas\_correctamente

Descripción: busca detectar un error al momento de leer los archivos.

CAMBIO.IN	CAMBIO.OUT(Esperado)	CAMBIO.OUT(Obtenido)
6 1 6 7 3 1 4 3 4 1 4 5 1 5 6 1 1 2 5 4 2 6 4 6 8	7 1	7 1

Informe:No presenta error.

Nombre: Caso\_03\_La distancia no es la correcta

Descripción: busca detectar un error en la distancia recorrida por el bus

CAMBIO.IN	CAMBIO.OUT(Esperado)	CAMBIO.OUT(Obtenido)
4 4 2 4 2 3 1 3 4 2	2 4	2 4

4 1 1 2 1 1		
----------------	--	--

Informe:No presenta error.

Nombre: Caso\_04\_Cambio\_direccion\_calles

Descripción: busca detectar cambios de direcciones en calles que no deberia

CAMBIO.IN	CAMBIO.OUT(Esperado)	CAMBIO.OUT(Obtenido)
10 1 9 13 1 4 2 4 5 1 5 8 3 8 9 1 9 10 1 4 3 2 3 2 2 3 6 4 2 7 2 8 7 1 6 8 1 8 10 10 1 2 5	7	7

Informe:No presenta error.

Nombre: Caso de estres

Se realizara una gran cantidad de operaciones para ver la eficacia del algoritmo

CAMBIO.IN	CAMBIO.OUT(Esperado)	CAMBIO.OUT(Esperado)
200 0 200 200 0 1 1 1 2 1 2 3 1 3 4 1 4 5 1 5 6 1 6 7 1 7 8 1 8 9 1 9 10 1 11 10 1 11 12 1 12 13 1 13 14 1 14 15 1	200 11 35 45 60 71 100	200 11 35 45 60 71 100

15 16 1		
16 17 1		
17 18 1		
18 19 1		
19 20 1		
20 21 1		
21 22 1		
22 23 1		
23 24 1		
24 25 1		
25 26 1		
26 27 1		
27 28 1		
28 29 1		
29 30 1		
30 31 1		
31 32 1		
32 33 1		
33 34 1		
35 34 1		
35 36 1		
36 37 1		
37 38 1		
38 39 1		
39 40 1		
40 41 1		
41 42 1		
42 43 1		
43 44 1		
45 44 1		
45 46 1		
46 47 1		
47 48 1		
48 49 1		
49 50 1		
50 51 1		
51 52 1		
52 53 1		
53 54 1		
54 55 1		
55 56 1		
56 57 1		
57 58 1		
58 59 1		
60 59 1		
60 61 1		
61 62 1		
62 63 1		
63 64 1		
64 65 1		
65 66 1		
66 67 1		
67 68 1		
68 69 1		
69 70 1		
71 70 1		
71 72 1		

72 73 1		
73 74 1		
74 75 1		
75 76 1		
76 77 1		
77 78 1		
78 79 1		
79 80 1		
80 81 1		
81 82 1		
82 83 1		
83 84 1		
84 85 1		
85 86 1		
86 87 1		
87 88 1		
88 89 1		
89 90 1		
90 91 1		
91 92 1		
92 93 1		
93 94 1		
94 95 1		
95 96 1		
96 97 1		
97 98 1		
98 99 1		
100 99 1		
100 101 1		
101 102 1		
102 103 1		
103 104 1		
104 105 1		
105 106 1		
106 107 1		
107 108 1		
108 109 1		
109 110 1		
110 111 1		
111 112 1		
112 113 1		
113 114 1		
114 115 1		
115 116 1		
116 117 1		
117 118 1		
118 119 1		
119 120 1		
120 121 1		
121 122 1		
122 123 1		
123 124 1		
124 125 1		
125 126 1		
126 127 1		
127 128 1		
128 129 1		

129 130 1		
130 131 1		
131 132 1		
132 133 1		
133 134 1		
134 135 1		
135 136 1		
136 137 1		
137 138 1		
138 139 1		
139 140 1		
140 141 1		
141 142 1		
142 143 1		
143 144 1		
144 145 1		
145 146 1		
146 147 1		
147 148 1		
148 149 1		
149 150 1		
150 151 1		
151 152 1		
152 153 1		
153 154 1		
154 155 1		
155 156 1		
156 157 1		
157 158 1		
158 159 1		
159 160 1		
160 161 1		
161 162 1		
162 163 1		
163 164 1		
164 165 1		
165 166 1		
166 167 1		
167 168 1		
168 169 1		
169 170 1		
170 171 1		
171 172 1		
172 173 1		
173 174 1		
174 175 1		
175 176 1		
176 177 1		
177 178 1		
178 179 1		
179 180 1		
180 181 1		
181 182 1		
182 183 1		
183 184 1		
184 185 1		
185 186 1		

186 187 1		
187 188 1		
188 189 1		
189 190 1		
190 191 1		
191 192 1		
192 193 1		
193 194 1		
194 195 1		
195 196 1		
196 197 1		
197 198 1		
198 199 1		
199 200 1		

Informe:No presenta error y el tiempo de ejecución fue rápido

La complejidad computacional del algoritmo de la clase “Bus” es de  $O(n \log n)$  por que la carga mas pesada de ejecución la realiza el algoritmo de Dijkstra.