**TRAFİK AKIŞININ MODELLENMESİ**

Arda Talu

(170202037) [arda.starkiler@gmail.com](mailto:arda.starkiler@gmail.com)

Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümü

**Özet:**

Projenin amacı kullanıcıdan alınan tüm yol yön bilgileri ve yollardan geçen ortalama araç sayıları bilgileri kullanılarak kullanıcı tarafından girilmemiş olan ortalama araç miktarının hesaplanmasıdır. Çözüme ulaşmak için ise Gauss-Jordan Eliminasyon Yöntemi araç olarak kullanılmasıdır.

**1.Temel Bilgiler**

Program yazılırken C dilinden ve görsel arayüz için <allegro.h> kütüphanesinden yararlanılmıştır. Kodları yazmak için ise Code::Blocks 17.12 IDE ve Notepad++ kullanılmıştır.

**2.Algoritma**

Program başlangıçta kullanıcıdan görsel yardımı ile betimlenmiş olan 2 harita(Harita 1 ve Harita 2) arasından hangisine göre işlem yapacağını seçmesini ister.

Seçtiği harita fark etmemeksizin haritaların giriş-çıkış yolları olan x,y,z,t yollarının hangilerinin giriş hangilerinin çıkış olduğunu sorar. Görsel üzerinde oklar yardımı ile hangi yolun ne yöne doğru olduğu gösterilir. Ara yollar olan Harita 1’de a,b,c,d yolları için ve Harita 2’de a,b,c,d,e yolları için x,y,z,t giriş-çıkış yolları aracılığı ile yön bilgisi girmesi istenir.

Ardından kullanıcıdan yollar için araç yoğunluğu değerlerinin girilmesi istenir. Alınan yol bilgileri gerekli denklem yolları ile matrise dökülür ve Gauss-Jordan Eliminasyon Yöntemi ile sonuç çıkartılır.

**3. Sözde Kod**

1. Başla
2. Oku(“Harita seciniz (Harita 1 icin 1 - Harita 2 icin 2)”)
3. Yaz(harita)
4. Eğer harita=1 ise
5. Oku(“Yollarin giris/cikis yonunu belirleyin. Giris icin 1, Cikis icin -1”)
6. Oku(“X yolu icin”)
7. Yaz(x)
8. Eğer x=1 ise
9. Oku(“X yolu giristir.”)
10. Eğer x=-1 ise
11. Oku(“X yolu cikistir.”)
12. Oku(“Y yolu icin”)
13. Yaz(y)
14. Eğer y=1 ise
15. Oku(“Y yolu giristir.”)
16. Eğer y=-1 ise
17. Oku(“Y yolu cikistir.”)
18. Oku(“Z yolu icin”)
19. Yaz(z)
20. Eğer z=1 ise
21. Oku(“Z yolu giristir.”)
22. Eğer z=-1 ise
23. Oku(“Z yolu cikistir.”)
24. Oku(“T yolu icin”)
25. Yaz(t)
26. Eğer t=1 ise
27. Oku(“T yolu giristir.”)
28. Eğer t=-1 ise
29. Oku(“T yolu cikistir.”)
30. Eğer(x=1 ve y=1 ve z=-1 ve t=-1)
31. Değilse Eğer(x=1 ve y=-1 ve z=1 ve t=-1)
32. Değilse Eğer(x=1 ve y=-1 ve z=-1 ve t=1)
33. Değilse Eğer(x=-1 ve y=1 ve z=1 ve t=-1)
34. Değilse Eğer(x=-1 ve y=1 ve z=-1 ve t=1)
35. Değilse Eğer(x=-1 ve y=-1 ve z=1 ve t=1)
36. Değilse
37. Yaz(“Hatali giris yaptiniz. Lutfen 2 giris 2 cikis giriniz.”)
38. Oku(“Ara caddelerin yonunu belirleyin:”)
39. Oku(“A yolu icin baslangic noktasi belirleyiniz. (x-t)”)
40. Yaz(aBas)
41. Oku(“A yolu icin yon belirleyiniz. (x-t)”)
42. Yaz(aYon)
43. Oku(“B yolu icin baslangic noktasi belirleyiniz. (x-y)”)
44. Yaz(bBas)
45. Oku(“B yolu icin yon belirleyiniz. (x-y)”)
46. Yaz(bYon)
47. Oku(“C yolu icin baslangic noktasi belirleyiniz. (z-t)”)
48. Yaz(cBas)
49. Oku(“C yolu icin yon belirleyiniz. (z-t)”)
50. Yaz(cYon)
51. Oku(“D yolu icin baslangic noktasi belirleyiniz. (y-z)”)
52. Yaz(dBas)
53. Oku(“D yolu icin yon belirleyiniz. (y-z)”)
54. Yaz(dYon)
55. Oku(“Yollarin arac yogunlugunu giriniz.”)
56. Oku(“X yolunun arac yogunlugunu giriniz.”)
57. Yaz(xYog)
58. Oku(“Y yolunun arac yogunlugunu giriniz.”)
59. Yaz(yYog)
60. Oku(“Z yolunun arac yogunlugunu giriniz.”)
61. Yaz(zYog)
62. Oku(“T yolunun arac yogunlugunu giriniz.”)
63. Yaz(tYog)
64. Oku(“A yolunun arac yogunlugunu giriniz.”)
65. Yaz(aYog)
66. Oku(“B yolunun arac yogunlugunu giriniz.”)
67. Yaz(bYog)
68. Oku(“C yolunun arac yogunlugunu giriniz.”)
69. Yaz(cYog)
70. Oku(“D yolunun arac yogunlugunu giriniz.”)
71. Yaz(dYog)
72. Değilse eğer harita=2 ise
73. Oku(“Yollarin giris/cikis yonunu belirleyin. Giris icin 1, Cikis icin -1”)
74. Oku(“X yolu icin”)
75. Yaz(x)
76. Eğer x=1 ise
77. Oku(“X yolu giristir.”)
78. Eğer x=-1 ise
79. Oku(“X yolu cikistir.”)
80. Oku(“Y yolu icin”)
81. Yaz(y)
82. Eğer y=1 ise
83. Oku(“Y yolu giristir.”)
84. Eğer y=-1 ise
85. Oku(“Y yolu cikistir.”)
86. Oku(“Z yolu icin”)
87. Yaz(z)
88. Eğer z=1 ise
89. Oku(“Z yolu giristir.”)
90. Eğer z=-1 ise
91. Oku(“Z yolu cikistir.”)
92. Oku(“T yolu icin”)
93. Yaz(t)
94. Eğer t=1 ise
95. Oku(“T yolu giristir.”)
96. Eğer t=-1 ise
97. Oku(“T yolu cikistir.”)
98. Eğer(x=1 ve y=1 ve z=-1 ve t=-1)
99. Değilse Eğer(x=1 ve y=-1 ve z=1 ve t=-1)
100. Değilse Eğer(x=1 ve y=-1 ve z=-1 ve t=1)
101. Değilse Eğer(x=-1 ve y=1 ve z=1 ve t=-1)
102. Değilse Eğer(x=-1 ve y=1 ve z=-1 ve t=1)
103. Değilse Eğer(x=-1 ve y=-1 ve z=1 ve t=1)
104. Değilse
105. Yaz(“Hatali giris yaptiniz. Lutfen 2 giris 2 cikis giriniz.”)
106. Oku(“Ara caddelerin yonunu belirleyin:”)
107. Oku(“A yolu icin baslangic noktasi belirleyiniz. (x-t)”)
108. Yaz(aBas)
109. Oku(“A yolu icin yon belirleyiniz. (x-t)”)
110. Yaz(aYon)
111. Oku(“B yolu icin baslangic noktasi belirleyiniz. (x-y)”)
112. Yaz(bBas)
113. Oku(“B yolu icin yon belirleyiniz. (x-y)”)
114. Yaz(bYon)
115. Oku(“C yolu icin baslangic noktasi belirleyiniz. (z-t)”)
116. Yaz(cBas)
117. Oku(“C yolu icin yon belirleyiniz. (z-t)”)
118. Yaz(cYon)
119. Oku(“D yolu icin baslangic noktasi belirleyiniz. (y-z)”)
120. Yaz(dBas)
121. Oku(“D yolu icin yon belirleyiniz. (y-z)”)
122. Yaz(dYon)
123. Oku(“E yolu icin baslangic noktasi belirleyiniz. (y-t)”)
124. Yaz(eBas)
125. Oku(“E yolu icin yon belirleyiniz. (y-t)”)
126. Yaz(eYon)
127. Oku(“Yollarin arac yogunlugunu giriniz.”)
128. Oku(“X yolunun arac yogunlugunu giriniz.”)
129. Yaz(xYog)
130. Oku(“Y yolunun arac yogunlugunu giriniz.”)
131. Yaz(yYog)
132. Oku(“Z yolunun arac yogunlugunu giriniz.”)
133. Yaz(zYog)
134. Oku(“T yolunun arac yogunlugunu giriniz.”)
135. Yaz(tYog)
136. Oku(“A yolunun arac yogunlugunu giriniz.”)
137. Yaz(aYog)
138. Oku(“B yolunun arac yogunlugunu giriniz.”)
139. Yaz(bYog)
140. Oku(“C yolunun arac yogunlugunu giriniz.”)
141. Yaz(cYog)
142. Oku(“D yolunun arac yogunlugunu giriniz.”)
143. Yaz(dYog)
144. Oku(“E yolunun arac yogunlugunu giriniz.”)
145. Yaz(eYog)
146. Değilse(Harita 1 veya Harita 2 Seçilmemişse)
147. Oku(“Hatali giris yaptiniz tekrar deneyin.”)
148. Dön(HaritaSecim)
149. Bitiş

**4.Kaynakça**

Using Allegro:

<https://liballeg.org/stabledocs/en/alleg000.html>

Structures and types defined by Allegro:

<https://liballeg.org/stabledocs/en/alleg001.html>

# Graphics modes

# <https://liballeg.org/stabledocs/en/alleg008.html>

# Bitmap objects

# <https://liballeg.org/stabledocs/en/alleg009.html>

# Loading image files

# <https://liballeg.org/stabledocs/en/alleg010.html>

### CODE (With back-substitution):

# <https://www.bragitoff.com/2018/02/gauss-elimination-c-program/>