



SAKARYA
UYGULAMALI BİLİMLER
ÜNİVERSİTESİ

VERİ YAPILARI ve ALGORİTMALAR

Proje Uygulaması

Uygulama Tarihi:
9. Hafta

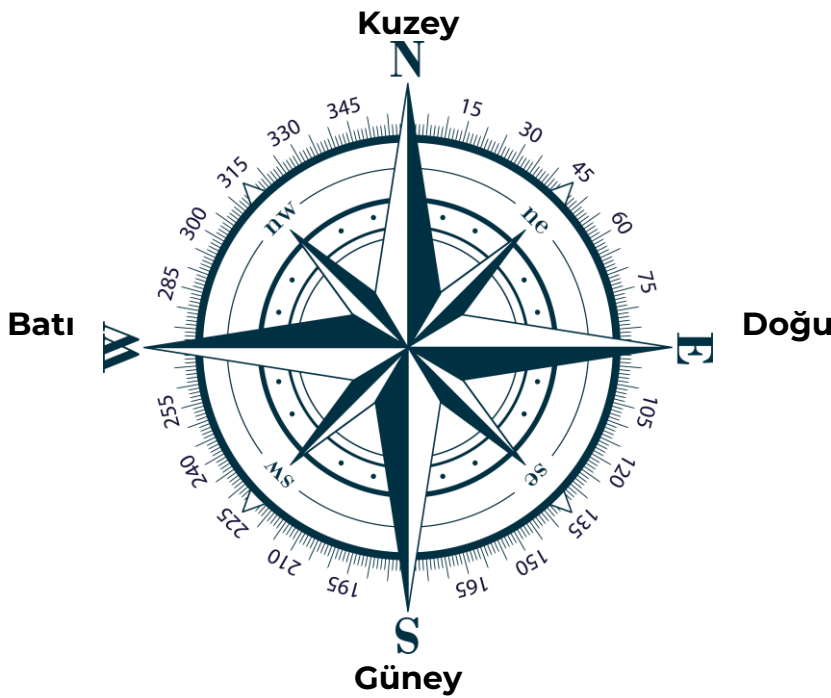
Dr.Öğr.Üyesi Selman HIZAL

Labirent (Maze) Oyunu: Backtraking Uygulaması

Yığıt (Stack) veri yapısı birçok problemin çözümünde kullanılan önemli bir veri yapısıdır. Örneğin bir web sayfasında gezinti yaptığımızda eski sayfalara geri dönebilmek için yığıt kullanılmaktadır. Ayrıca birçok yazılımda (Word, Excel, Photoshop vb.) yaptığınız işlemleri geri veya ileri alabilmeniz için seçenekler bulunmaktadır. Bazı oyunlarda yaptığımız hamleleri yığıt kullanarak defalarca geri alabiliyoruz.

Problem: $M \times N$ boyutlarında bir alanda bulunan labirentin başlangıç noktası ve çıkış noktası arasında yollar kullanılarak gezinti sağlanması ve çıkışın bulunması istenmektedir. Labirent içerisinde gezinirken ana coğrafi yönler kullanılacaktır.

Coğrafi yönler, dünya üzerindeki yer ve varlıkların konumunu tanımlamakta kullanılan, dünyanın kutupları baz alınarak oluşturulmuş terimlerdir. Başlıca dört coğrafi yön vardır: kuzey, güney, doğu ve batı. Bunlara ana yönler denir. Bir kişi yüzünü kuzeye dönerse arkası güney, sağ tarafı doğu ve sol tarafı batıdır.



Labirentin YÜKSEKLİK ve GENİŞLİK bilgilerini kodunuz içerisinde aşağıdaki şekilde belirtebilirsiniz.

```
#define YUKSEKLIK 20
#define GENISLIK 50
```

Labirent için ise bir text dokümanı içerisinde aşağıdaki örnekteki gibi bir yapı oluşturabilirsiniz. Burada # işareti ile labirentinizin içerisindeki duvarları tanımlamış olursunuz. Boşluklar arasından geçerek SAĞ, SOL, AŞAĞI ve YUKARI olacak şekilde ilerlenebilmektedir.

```
typedef enum {
    ASAGI, SOL, YUKARI, SAG
} Yon;
```

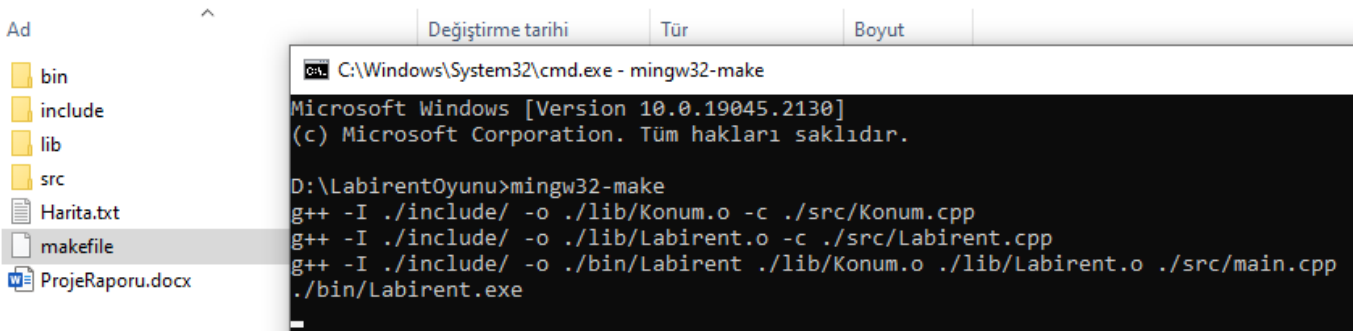
Programınız Harita.txt dosyasından labirenti `FILE* fp = fopen("Harita.txt", "r");` ile okuyup; iki boyutlu bir karakter dizisine `char harita[YUKSEKLIK][GENISLIK];` aktararak, başlangıç ve bitiş noktası arasında gezintiye başlayabilirsiniz.

Programınız farklı labirent Haritaları verildiğinde de çalışabilmelidir. Aşağıda örnek Haritalar verilmiştir.

Harita.txt		Boyut(20,50)
<pre> 1 ##### 2 # 3 ##### 4 ##### 5 # 6 ##### 7 ##### 8 ##### 9 ##### 10 ##### 11 ##### 12 ##### 13 ##### 14 # 15 ##### 16 ##### 17 ##### 18 ##### 19 ##### 20 #####</pre>		<p>Başlangıç: Konum(0,19)</p> <p>Çıkış: Konum(19,6)</p>

	<p>Boyut(20,50)</p> <p>Başlangıç: Konum(0,19)</p> <p>Çıkış: Konum(19, 6)</p>
--	--

Yazmış olduğunuz programınız bir Makefile dosyası ile aşağıdaki gibi mingw32-make komutu verildiğinde derlenerek çalıştırılabilir olması gerekmektedir. Ayrıca klasör yapınızda şekildeki gibi olmalıdır.



Programınız derlendikten sonra /bin/Labirent.exe programı çağrılarak aşağıdaki gibi çalışacaktır.

```
C:\Windows\System32\cmd.exe - mingw32-make
```

[illegible]

Yönünüzü gösteren karakteri aşağıdaki gibi içi dolu oklar olacak şekilde tercih edebilirsiniz.

```
char YonChr[] = {31, 17, 30, 16}; // ASAGI, SOLA, YUKARI, SAG
```

Ctrl	Dec	Hex	Char	Code
A	1	01	☺	SOH
B	2	02	☹	STX
C	3	03	♥	ETX
D	4	04	♦	EOT
E	5	05	♣	ENQ
F	6	06	♠	ACK
G	7	07	•	BEL
H	8	08	◼	BS
I	9	09	↻	HT
J	10	0A	◻	LF
K	11	0B	♂	VT
L	12	0C	♀	FF
M	13	0D	♪	CR
N	14	0E	♫	SO
O	15	0F	⚙	SI

Ctrl	Dec	Hex	Char	Code
P	16	10	▶	DLE
Q	17	11	◀	DC1
R	18	12	↕	DC2
S	19	13		DC3
T	20	14	¶	DC4
U	21	15	§	NAK
V	22	16	▬	SYN
W	23	17	↕	ETB
X	24	18	↑	CAN
Y	25	19	↓	EM
Z	26	1A	→	SUB
[27	1B	←	ESC
\	28	1C	L	FS
]	29	1D	↔	GS
^	30	1E	▲	RS
_	31	1F	▼	US

Projenizin mutlaka rapor sayfası bulunmalıdır. Kendinizi, algoritmanızı ve programınızı tanıtmamız ve hangi kaynaklardan faydalandığınızı göstermeniz gerekmektedir.