
BİL-113 Bilgisayar Programlama - I

Ödev - 2

Veriliş Tarihi: 11.10.2010

Teslim Tarihi: 25.10.2010 (Saat : 23:59)

Teslim Şekli: AdSoyadOdev2.rar (veya zip) dosyasına ödevinizi paketleyip, kdursun@etu.edu.tr adresine konusu tam olarak "[bil113] odev2" olacak şekilde e-posta'ya dosyayı ekleyerek gönderiniz.

Kurallar: Geç gönderilen ödevler kabul edilmez. Kopya kesinlikle yasaktır, kopya veren ve alan öğrenciler ödevden 0 alırlar ve ayrıca üniversite disiplin yönetmeliği kuralları bu öğrencilere uygulanır.

Av avcı problemi (hunter-prey problem) yapay zeka araştırmalarındaki önemli problemlerden biridir. Bu problem, yeni geliştirilen bir planlama algoritması için bir performans göstergesi (benchmark) olarak yaygın biçimde kullanılır. Planlama, avın avcıyı en kısa sürede yakalaması için yapılır. Tabi problemin bir çok farklı versiyonuna rastlamak mümkündür. Problemin en yaygın kullanıldığı tip, bir grid (ızgara) üzerinde tanımlanandır. Grid'i nxn'lik bir satranç tahtası gibi düşünebilirsiniz. Grid'in her iki boyutu da 1'den n'e kadar numaralanmıştır ve (x,y) koordinatları şeklinde konum bildirmek için kullanılırlar.

Bu ödevde, bir grid'deki av-avcı problemini modelleyen bir Java programı yazacaksınız. Aşağıda grid ve üzerindeki bir av (P: Prey) ve avcı (H: Hunter) gösteriliyor. '.' karakteri boş alanları ifade ediyor.

	1	2	3	4	5	6	7	8	9	10
1
2
3	P	.	.	.
4
5
6	.	.	H
7
8
9
10

Yukarıdaki örnek gridin boyutu 10x10. Avcının (H) koordinatları (3,6), avın (P) koordinatları ise (7,3). (Grid'in koordinatları aynen şekilde gösterildiği gibi kodlanmalıdır!) Amaç, adından da anlaşılacağı gibi avı en kısa zamanda yakalamak. Fakat problemimizde av ve avcının belli kabiliyet farklılıkları olacak.

Hareket: Av, her adımda, grid'in dışına çıkmamak koşuluyla yukarı,aşağı,sağa ve sola hareket edebilir. Avcı ise avın hareketlerini yapabilirken buna ek olarak bir de çapraz hareket edebiliyor. Buna göre yukarıdaki gridde, bir sonraki adımda, avın hareket edebileceği noktalar, (7,2), (7,4), (6,3), (8,3), avcının hareket edebileceği noktalarsa (3,7), (3,5), (2,6), (4,6), (4,5), (2,5), (2,7) ve (4,7).

Görüş: Av ve avcının hareket kabiliyeti dışında, görüş kabiliyetleri de farklı. Buna göre, avcı avı her mesafeden görebilirken, av, avcıyı sadece aralarındaki mesafe 1.5 birimin altına inerse görebiliyor. Buradaki uzaklık hesabı da Euclid uzaklığı ($\sqrt{(x1 - x2)^2 + (y1 - y2)^2}$) olarak hesaplanıyor. Buna göre, örneğin, yukarıdaki grid'deki av ve avcının uzaklıkları:

$$\sqrt{(3 - 7)^2 + (6 - 3)^2} = 5$$

Avın hareket politikası programınızda şu şekilde olmalı: Av, avcıyı görmediği sürece hareketsiz kalacak, farketmediği anda ise hareket edebileceği noktalar içinde avcıdan en uzak olan noktaya kaçacaktır. Birden fazla böyle nokta varsa onlardan herhangi birine hareket edebilir.

Avcının hareket politikasını siz belirleyeceksiniz. Avcıyı yapabildirdiğiniz kadar zekice hareket ettirmelisiniz. Yani, avı yakalayabildiğiniz kadar kısa zamanda yakalamaya çalışmalısınız. Ödev değerlendirilirken avcının yakalama süresi dikkate alınacaktır.

Program ilk çalıştığında, grid'in boyutlarıyla, avın ve avcının yerini kullanıcıdan almalıdır. Bunlar alındıktan sonra kullanıcıdan programın modu (çalışma şekli) alınmalıdır. Programın modları aşağıda açıklanmıştır.

Programın modları: Yazacağınız kodun iki modu (çalışma şekli) olmalı. Bunlardan ilki (interaktif mod), her adımda gridin yeni durumunu çizdikten sonra devam etmek için kullanıcıdan input bekleyecektir. Kullanıcı 'e' harfi girdiği sürece simülasyon devam etmelidir. Bu modda 'e' den farklı bir input girilirse, program direk olarak çıkmalıdır. Avcı avı yakalayınca, kaç adımda avın yakalandığı ile ilgili bir çıktı verilip, çıkılır. Aşağıda interaktif mod için örnek bir çalışma gösterilmektedir.

Grid boyutlarını giriniz : 9 9
Avın yerini giriniz : 7 3
Avcının yerini giriniz : 4 5
Program modu (i/s) : i

	1	2	3	4	5	6	7	8	9
1
2
3	P	.	.
4
5	.	.	.	H
6
7
8
9

Devam? e

	1	2	3	4	5	6	7	8	9
1
2
3	P	.	.
4	H
5
6
7
8
9

Devam? e

	1	2	3	4	5	6	7	8	9
1
2
3	H	P	.	.
4
5
6
7
8
9

Devam? e

	1	2	3	4	5	6	7	8	9
1
2
3	H	P	.
4
5
6
7
8
9

Devam? e

	1	2	3	4	5	6	7	8	9
1
2
3	H	P
4
5
6
7
8
9

Devam? e

	1	2	3	4	5	6	7	8	9
1
2
3
4	#
5
6
7
8
9

Avcı, avı 5 adımda yakaladı.

Programın ikinci modu simülasyon modu olmalıdır. Bu modda aradaki adımlar ekrana çizilmeden, kullanıcıdan da herhangi bir input beklemeden avcı avı yakalayana dek program çalışmalıdır. Bu modda, sadece ilk ve son durumlar ekrana çizilmelidir. Aşağıda simülasyon modu için bir örnek çalışma gösteriliyor.

Grid boyutlarını giriniz : 9 9
Avın yerini giriniz : 7 3
Avcının yerini giriniz : 4 5
Program modu (i/s) : s

	1	2	3	4	5	6	7	8	9
1
2
3	P	.	.
4
5	.	.	.	H
6
7
8
9

	1	2	3	4	5	6	7	8	9
1
2
3
4	#
5
6
7
8
9

Avcı, avı 5 adımda yakaladı.