

DATA STRUCTURES

Proje

Asansörlü ve Merdiven Alternatifli Otelde Kuyruk Simülasyonu

Ad-soyad, Numara

Mehmet Anıl Taysi, 05170000022

Emel Kayacı, 05180000087

İÇİNDEKİLER

1. Program 1

1.1 Gerçekleştirilen Platform ve Dil ve Sürüm Adı.....	3
1.2 Programın kısa tanımı.....	3
1.3 Program için kullanılan veri yapılarının, sınıfların ve metotların açıklamaları.....	3
1.4 Örnek sonuçlar.....	7
1.5 Yazılım geliştirme için harcanan süreler.....	12

1. Program

1.1 Gerçekleştirilen Platform ve Dil ve Sürüm Adı

Gerçekleştirilen platform: Visual Studio 8.3.10 Dil: C#

1.2 Programın kısa tanımı

Bizden 2 adet asansörü ve 1 adet merdiveni bulunan bir otelde turistlerin katlara çıkış süreleri hakkında çeşitli hesaplamaları içeren programın yazılması istenmektedir. Asansörlerden biri turistleri FIFO düzeninde almakta olup 4 saniyede 1 kat çıkmakta iken diğeri turistleri PQ düzeninde alıp 1.66 saniyede 1 kat çıkmaktadır. Merdiven sadece random olarak üretilen veri seti üzerinden hesaplamaların birinde kullanılmakta olup turistler merdivende 9 saniyede 1 kat çıkabilmektedirler. Her bir asansör durduğu katlarda 5 saniye bekleme yapıp 4'er kişi alabilmektedir. Ayrıca asansörlerin çıkış ve iniş hızları aynıdır.

Projede iki tür veri seti kullanılmakta olup proje dökümanında verilen veri seti üzerinden a, b ve c maddelerindeki hesaplama ve listelemelerin, random olarak üretilen (1 ile 50 kat arasında 43 adet kişi) veri seti üzerinden ise d, e ve f maddelerinin kodlanması istenmektedir.

Hesaplamalar her bir turist için İTS (işlem tamamlama süresi) ve OİTS (ortalama işlem tamamlama süresi)'yi ve ortalama olarak her bir turistin en kısa sürede işlemini tamamlayabilmesi için hangi asansöre yönlendirilmesi gerektiğini içermektedir. Bu yönlendirme işleminde her bir olasılığın denenmesi yerine bilgisayarlı işlem yükünden kurtarmak ve zamandan tasarruf etmek için genetik algoritma optimizasyon yöntemi kullanılmıştır.

1.3 Program için kullanılan veri yapılarının, sınıfların ve metotların açıklamaları

- **Program için kullanılan veri yapıları ve açıklamaları:**
 - **Queue:** Kuyruk sınıfında bulunması gereken enqueue, dequeue gibi metotlar kodlanmış olup ayrıca kuyrukta bulunan eleman sayısını döndüren Count metodu kullanılmıştır. Bu projede kuyruk veri yapısı kullanılmasının nedeni turistlerden sıraya ilk girenlerin sıradan ilk çıkıp asansöre gönderilmesindendir.
 - **Priority Queue:** Bu projede priority queue kullanılmasının nedeni asansörü öncelikle düşük katlara çıkacak turistlere kullandırmaktır. Bu turistler işlerini kısa bir zamanda bitirip daha sonra üst katlara çıkacak turistler asansörlere alınmaya başlanacaktır. Bunun için turistler kat numaraları artan sırada asansörler önünde kuyruk oluşturup kuyrukta ilk sırada bulunan sıradan ilk ayrılan birey olup asansöre gönderilecektir.
 - **HashSet:** Bireylerin çıkacakları kat numaraları sırasıyla 1, 3, 8, 8 olsun. Bu durumda 8. Kata çıkan her iki bireyin de its süreleri aynı olacağından tekrar bir hesaplama yapmaya gerek yoktur. Bu nedenle HashSet kullanılmıştır. Asansördeki kişilerin kat numaraları kümeye atılarak kümeye aynı değer iki kez eklenemediğinden bu durumda küme veri yapısının kullanılması kolaylık sağlamıştır.
 - **Array:** Turistlerin kullandıkları ulaşım araçları (0: FIFO, 1: PQ, 2: Merdiven) tek boyutlu dizide tutulmuştur.
 - **Generic list:** Turist ve DNA nesnelerinin tutulması için generic list kullanılmıştır.
- **Program için kullanılan sınıf ve açıklamaları:**
 - **Program sınıfı:** Main metodunu ve programın çalışması için gereken ana kodu içeren sınıftır.
 - **Queue sınıfı:** Queue veri yapısını kullanabilmemizi sağlayan sınıftır.

- **AsansörPQ sınıfı:** AsansörPQ, Queue sınıfının bir alt sınıfı olup Dequeue ve Count metotlarını Queue sınıfından kullanır. Enqueue yapısı önceliğe göre olduğundan kendine özgü enqueue metodu yazılmıştır. Turistlerin gruplar şeklinde (asansörün kapasitesine eşit veya daha az) PQ asansöre alınıp its değerlerinin hesaplanması bu sınıfta gerçekleşir.
- **AsansörFIFO sınıfı:** AsansörFIFO, Queue sınıfının bir alt sınıfı olup Dequeue ve Count metotlarını Queue sınıfından kullanır. Turistlerin gruplar şeklinde (asansörün kapasitesine eşit veya daha az) FIFO asansöre alınıp its değerlerinin hesaplanması bu sınıfta gerçekleşir.
- **Turist sınıfı:** Turist nesnesi oluşturmamızı sağlar.
- **GenetikAlgoritma sınıfı:** Genetik algoritma optimizasyon yönteminin çalışabilmesi için gereken metotları bulundurur.
- **DNA sınıfı:** Genetik algortmada kullanılan popülasyon listesi DNA nesnelerinden oluşmaktadır. Her bir generasyonda 10 adet DNA bulunup bu DNA'lar turistlerin ulaşım seçeneklerini (0: FIFO, 1: PQ, 2: Merdiven) ve oits süresini tutar.
- **Program için kullanılan metot ve açıklamaları:**
 - **Program sınıfı içerisinde kullanılan metotlar**
 - **Main metodu:** Kullanıcı menüsünü içerir, menü içerisinde proje dökümanında belirtilen maddelere göre uygun metotları çağırır.
 - **enUygunAsansörYazdır metodu:** Genetik algoritma optimizasyon yöntemi sonucunda elde edilen minimum oits değerini oluşturan kişilerin hangi asansörlere yerleştirildiğini numaralarıyla birlikte ekrana yazdırır.
 - **oitsKazançYazdır metodu:** Turistleri en uygun asansörlere yönlendirme işleminde elde edilen minimum oits değerine göre sadece AsansörPQ'yu ve sadece AsansörFIFO'yu kullanma durumlarına göre kaçar saniye kazanç sağladığını ekrana yazdırır.
 - **oitsYazdır metodu:** B maddesinde istenen oits süresini ekrana yazdırır.
 - **oitsToplamBul metodu:** Her bir kişinin its değerlerini toplayarak oits değeri için gereken toplam değerini bulur. Bu değer daha sonra metot dışında kullanıldığında kişi sayısına bölünerek oits değerine ulaşılmış olur.
 - **oitsRandomToplamBul metodu:** Her bir kişinin her iki asansörden birisinin sırasına random geçtiklerindeki its değerlerini toplayarak oits değeri için gereken toplam değerini bulur. Bu değer daha sonra metot dışında kullanıldığında kişi sayısına bölünerek oits değerine ulaşılmış olur.
 - **asansörlereGönder metodu:** Turistleri sadece FIFO, sadece PQ ve random düzendeki asansörlere göndererek bu asansörlerde her bir kişi için çıkış süresi hesaplanır.
 - **itsYazdır metodu:** B maddesinde istenen its süresini kişilerin bilgileriyle beraber

ekrana yazdırır.

- **tabloYazdır metodu:** B maddesinde istenen süre kazanç tablosunu ekrana yazdırır.
- **deneySetiOluştur metodu:** Genetik algorithmada kullanılacak deney setini (ilk popülasyonu) oluşturur.
- **MerdivenSüreHesapla metodu:** F maddesinde merdiven ile katlarına çıkacak olan her bir kişinin çıkış süresini hesaplar ve merdiven ile çıkan tüm kişilerin toplam süresini bulur.
- **GenetikAlgoritmaÇalıştır metodu:** Her bir generasyon için yeni popülasyon oluşturur. Generasyonda oits süresi en az olan üç birey yeni popülasyona değiştirilmeden aktarılırken geriye kalan yedi birey genetik algoritma sonucunda elde edilmiş çocuklardan oluşur. Generasyon sayısı kadar tekrar eden for döngüsünün dışında ise son popülasyondaki minimum oits değerine sahip eleman döndürülür.
- **ÇocukAsansörAta metodu:** Genetik algoritma sonucunda crossover ve mutate metodlarıyla oluşturulmuş çocukları uygun asansörlere gönderebilmek için iki asansör ve merdiven için listeler yaratılıp çocuklar uygun asansör sırasına eklenir. Daha sonra **toplamSüreBul metodu** ile bu listelerdeki değerler asansörlere alınıp her bir kişi için its hesaplanır. Ayrıca oits hesabı için gereken toplam süre hesaplanır.
- **RandomSırayaYerleştir metodu:** Turistleri random asansör, merdiven sırasına yerleştirir. Merdiven sadece f maddesinde kullanıldığından kod tekrarına düşülmemesi adına bool tipinde merdiven_kontrol değişkeni tanımlanmıştır.
- **asansörFIFOyaAl metodu:** Turistleri FIFO düzenindeki asansöre alır. Döngü sayısı değişkeni 4'erli kaç grup olduğunu belirtir.
- **asansörPQyaAl metodu:** Turistleri PQ düzenindeki asansöre alır. Döngü sayısı değişkeni 4'erli kaç grup olduğunu belirtir. PQ asansörün hızı f şıkında değiştiğinden kod tekrarına düşmemek adına merdiven_kontrol değişkeni tanımlanmış olup merdiven açıldığı takdirde PQ asansörün hızını iki katına çıkarır.
- **turistOluştur metodu:** Turistlere numara, isim ve kat numarası verir.
- **nameGenerator metodu:** Turistlere random isim verir.
- **Queue sınıfı içerisinde kullanılan metotlar**
 - **Count metodu:** Kuyruk içerisinde bulunan eleman sayısını döndürür.
 - **Dequeue metodu:** Kuyruktan eleman siler.
 - **Enqueue metodu:** Kuyruğa eleman ekler.
- **AsansörPQ sınıfı içerisinde kullanılan metotlar**
 - **Enqueue metodu:** Öncelik sırasına göre, kat numarası düşük olan bireyleri, kuyruğa ekler.

- **asansöreEkle metodu:** Asansörün kapasitesine (4 kişi) eşit veya düşük olan grupları PQ düzenindeki asansöre alır. Kişiler PQ asansör kuyruğuna önceliklerine göre alındıklarından dequeue ile kuyruktan çıkarıldıklarında önce kat numarası düşük bireyler asansöre bindirilmiş olur.
 - **süreHesapla metodu:** Her bir asansör kişi grubundaki (asansörün kapasitesine eşit veya daha az sayıda kişi içeren grup) her bir kişinin PQ asansörde its hesaplar. Ayrıca gruptaki tüm bireyler kata çıkıp asansör boşaldıktan sonra iniş süresini de ekleyerek bir sonraki grup ile gelecek olan bireylerin bekleme süresine asansörün iniş süresini ve kendilerinden önceki grupların toplam süresini ekler.
 - **PQsüre metodu:** Yukarıda açıklaması verilmiş olan asansöreEkle ve süreHesapla metodları private tanımlanmış olup her iki metodun Main sınıfı içerisinde çağırılabilmesi için PQsüre metodu yazılmıştır. Bu metod içerisinde asansör grubu önce PQ düzenindeki asansöre eklenir, daha sonra süreHesapla metodu ile her bir kişinin PQ asansörde its hesaplanır.
- **AsansörFIFO sınıfı içerisinde kullanılan metotlar**
- **asansöreEkle metodu:** Asansörün kapasitesine (4 kişi) eşit veya düşük olan grupları FIFO düzenindeki asansöre alır. Örneğin bu dört kişilik gruptaki bireylerin çıkacakları kat numaraları sırasıyla 3, 8, 8, 1 olsun. Asansör ilk 1 sonra 3 daha sonra da 8. Kata çıkacağından burada bir PQ düzeni bulunmaktadır. Bu nedenle asansör grubu için program içerisinde daha önce yazılmış olan PQ veri yapısı kullanılmıştır.
 - **süreHesapla metodu:** Her bir asansör kişi grubundaki (asansörün kapasitesine eşit veya daha az sayıda kişi içeren grup) her bir kişinin FIFO asansörde its hesaplar. Ayrıca gruptaki tüm bireyler kata çıkıp asansör boşaldıktan sonra iniş süresini de ekleyerek bir sonraki grup ile gelecek olan bireylerin bekleme süresine asansörün iniş süresini ve kendilerinden önceki grupların toplam süresini ekler.
 - **FIFOsüre metodu:** Yukarıda açıklaması verilmiş olan asansöreEkle ve süreHesapla metodları private tanımlanmış olup her iki metodun Main sınıfı içerisinde çağırılabilmesi için FIFOsüre metodu yazılmıştır. Bu metod içerisinde asansör grubu önce FIFO düzenindeki asansöre eklenir, daha sonra süreHesapla metodu ile her bir kişinin FIFO asansörde its hesaplanır.
- **GenetikAlgoritma sınıfı içerisinde kullanılan metotlar**
- Ek:** int[] child, int[] parent1, int[] parent2 dizilerinin boyutu turist sayısı kadar olup bu diziler indeks değerlerinde FIFO asansör 0, PQ asansör 1 ve merdiven 2 olmak üzere her bir turistin hangi ulaşım aracını kullandığını tutar.
- **ÇocukBelirle metodu:** Crossover metodunu çağırarak iki parent dizisinin çaprazlanması sonucunda oluşan çocuğu döndürür.
 - **RouletteWheelSelection metodu:** Parent seçimi için kullanılan seçim yöntemidir. totalSumPopulation değişkeni her bir popülasyonun oits değerini toplar. Daha sonrasında oits değerine rulet tekerleğinde daha küçük bir dilim ayrılacak şekilde partialSum değerleri toplanır.
 - Bu kod tekerlekten ziyade yüzdelik dilimlere ayrılmış sütun gibi düşünüldüğü kodlanmıştır. Sütun 100 birim olarak düşünülmüş olup her bir popülasyonun partialSum değeri yüzdelik dilimini göstermektedir. (Oits büyük olanın yüzdelik dilimi

daha düşüktür.) Böylece random olarak sütun birimleri arasında değer belirlendiğinde yüzdelik dilimi yüksek (oits değeri düşük) olan popülasyondaki bireyin seçilme olasılığı daha yüksek olmaktadır.

- **Crossover metodu:** Parent dizilerini çarpazlar. Midpoint belirleyerek parent1 dizisinin midpoint değerine kadar olan kısmını, parent2 dizisinin ise midpoint değerinden sonraki olan kısmını alır. Ayrıca Mutate metodu çağırılıp mutasyonun gerçekleşebilme olasılığının değeri burada belirlenir.
- **Mutate metodu:** Çocuk dizisinde mutasyon gerçekleştirir. Rastgele bir indeksteki değeri değiştirir. (Aynı da kalabilir.)
- **ParentBelirle metodu:** Main metodunda kullanılan metot olup hem parent seçimi hem de parent dizilerinin çarpazlanması sonucu oluşan çocuk uygun metotlar çağırılarak belirlenir.

1.4 Örnek sonuçlar

b.1) Turistler sadece AsansörFIFO'ya, sadece AsansörPQ'ya ve her iki asansörden birisinin sırasına random binebildiğinde her bir turistin İTS ve OİTS değerlerinin tablosu

ITS VE OİTS DEĞERLERİ TABLOSU						
Numara	İsim	Kat Numarası	Sadece FIFO ITS	Sadece PQ ITS	Random ITS	
0	Bkcvx	3	17	31.62	22	
1	yewGb	8	47	322.68	104.8	
2	nzQVt	8	47	322.68	52	
3	ZpFEM	4	26	64.88	31	
4	GaSHk	4	110	64.88	41.6	
5	EPVcC	1	88	6.66	9	
6	gXIBh	1	88	6.66	6.66	
7	EgKcj	3	101	31.62	19.98	
8	qfCyI	2	139	13.32	13.32	
9	QtADp	5	156	71.54	48.26	
10	sFBwY	6	165	148.04	118	
11	aRVKs	8	178	322.68	104.8	
12	iIuEv	1	219	6.66	93	
13	QrPop	5	240	71.54	48.26	
14	tfCUV	7	258	154.7	98.14	
15	bYUyT	6	249	148.04	118	
16	LPSmJ	3	308	31.62	34.94	
17	aVLcG	7	329	154.7	127	
18	nKzjA	2	299	24.96	13.32	
19	hKxDs	7	329	316.02	188	
FIFO OİTS: 169.65						
PQ OİTS: 115.78						
RANDOM OİTS: 64.60						

b.2) Asansörlere göre süre kazançlarını gösteren tablo

KAZANÇ TABLOSU		
Numara	PQ'nun FIFO'ya göre Süre Kazancı	Random durumun FIFO'ya göre Süre Kazancı
4	45.12 saniye	68.4 saniye
5	81.34 saniye	79 saniye
6	81.34 saniye	81.34 saniye
7	69.38 saniye	81.02 saniye
8	125.68 saniye	125.68 saniye
9	84.46 saniye	107.74 saniye
10	16.96 saniye	47 saniye
11		73.2 saniye
12	212.34 saniye	126 saniye
13	168.46 saniye	191.74 saniye
14	103.3 saniye	159.86 saniye
15	100.96 saniye	131 saniye
16	276.38 saniye	273.06 saniye
17	174.3 saniye	202 saniye
18	274.04 saniye	285.68 saniye
19	12.98 saniye	141 saniye

c) En uygun asansör belirleme (20 kişilik test verisi için)

EN UYGUN ASANSÖRLER	
0 numaralı kişi:	FIFO
1 numaralı kişi:	PQ
2 numaralı kişi:	PQ
3 numaralı kişi:	FIFO
4 numaralı kişi:	PQ
5 numaralı kişi:	PQ
6 numaralı kişi:	PQ
7 numaralı kişi:	PQ
8 numaralı kişi:	FIFO
9 numaralı kişi:	FIFO
10 numaralı kişi:	FIFO
11 numaralı kişi:	PQ
12 numaralı kişi:	PQ
13 numaralı kişi:	FIFO
14 numaralı kişi:	FIFO
15 numaralı kişi:	FIFO
16 numaralı kişi:	PQ
17 numaralı kişi:	PQ
18 numaralı kişi:	PQ
19 numaralı kişi:	PQ
Minimum OITS: 50.88	

d) 1 ile 50 kat arasında random kata çıkacak 43 adet kişi için b seçeneğindeki deneyler ve listeleme

ITS VE OITS DEĞERLERİ TABLOSU

Numara	İsim	Kat Numarası	Sadece FIFO ITS	Sadece PQ ITS	Random ITS
0	XEoNm	25	115	1007.14	115
1	QieCf	18	82	465.36	82
2	zRKRB	49	216	17294.94	216
3	UYwUo	1	9	6.66	9
4	orcRa	33	554	2098.98	463.58
5	BMThX	1	421	6.66	6.66
6	iZvYc	1	421	6.66	6.66
7	sqvSP	33	554	2098.98	463.58
8	KNVGA	31	830	1022.1	551
9	RIrne	16	760	216.08	481
10	mLPJD	14	747	99.76	78.12
11	gcXmI	17	769	222.74	490
12	EPxfw	10	1004	83.12	66.48
13	ZVmoJ	20	1049	478.68	197.76
14	dceFe	9	995	33.26	24.94
15	bVVQJ	9	995	33.26	59.82
16	shgCl	32	1272	2092.32	560
17	xKmUw	33	1281	2098.98	976.94
18	QTigK	16	1203	216.08	181.12
19	vPqKa	9	1170	33.26	729
20	wWqdA	45	1608	17278.3	878
21	JHJec	15	1483	209.42	84.78
22	KGGdt	1	1422	6.66	6.66
23	Xiutz	47	1621	17286.62	891
24	PiRrt	31	1948	1022.1	455.26
25	KTxuj	9	1855	76.46	729
26	LfLpM	5	1834	21.62	1104
27	VqCCb	31	1948	1022.1	455.26
28	qAism	44	2268	8606.8	2071.8
29	JMfnn	42	2255	8598.48	1262
30	MVwKK	24	2173	490.32	209.4
31	OEykq	37	2230	4269.38	988.58
32	pJOuQ	50	2664	34667.88	1299
33	fvQQX	34	2590	4254.4	1225
34	ggrSJ	19	2525	472.02	191.1
35	PaYFj	47	2647	17286.62	2081.78
36	xoJHX	40	3039	8590.16	998.56
37	piAey	50	3084	34667.88	2091.76
38	QrSLm	11	2913	89.78	1548
39	gLNbH	33	3006	4247.74	1641
40	OwOQG	35	3429	4261.06	1654
41	fdeBv	49	3495	34661.22	1715
42	ngtAt	42	3462	8598.48	1006.88

FIFO OITS: 1,626.65

PQ OITS: 5,588.38

RANDOM OITS: 705.73

KAZANÇ TABLOSU

Numara	PQ'nun FIFO'ya göre Süre Kazancı	Random durumun FIFO'ya göre Süre Kazancı
3	2.34 saniye	
4		90.4200000000001 saniye
5	414.34 saniye	414.34 saniye
6	414.34 saniye	414.34 saniye
7		90.4200000000001 saniye
8		279 saniye
9	543.92 saniye	279 saniye
10	647.24 saniye	668.88 saniye
11	546.26 saniye	279 saniye
12	920.88 saniye	937.52 saniye
13	570.32 saniye	851.24 saniye
14	961.74 saniye	970.06 saniye
15	961.74 saniye	935.18 saniye
16		712 saniye
17		304.06 saniye
18	986.92 saniye	1021.88 saniye
19	1136.74 saniye	441 saniye
20		730 saniye
21	1273.58 saniye	1398.22 saniye
22	1415.34 saniye	1415.34 saniye
23		730 saniye
24	925.9 saniye	1492.74 saniye
25	1778.54 saniye	1126 saniye
26	1812.38 saniye	730 saniye
27	925.9 saniye	1492.74 saniye
28		196.2 saniye
29		993 saniye
30	1682.68 saniye	1963.6 saniye
31		1241.42 saniye
32		1365 saniye
33		1365 saniye
34	2052.98 saniye	2333.9 saniye
35		565.22 saniye
36		2040.44 saniye
37		992.24 saniye
38	2823.22 saniye	1365 saniye
39		1365 saniye
40		1775 saniye
41		1780 saniye
42		2455.12 saniye

e) d seçeneğinde üretilen 43 adet kişi için en uygun asansöre belirleme

EN UYGUN ASANSÖRLER

0 numaralı kişi: PQ
1 numaralı kişi: FIFO
2 numaralı kişi: PQ
3 numaralı kişi: PQ
4 numaralı kişi: PQ
5 numaralı kişi: PQ
6 numaralı kişi: PQ
7 numaralı kişi: PQ
8 numaralı kişi: PQ
9 numaralı kişi: FIFO
10 numaralı kişi: PQ
11 numaralı kişi: FIFO
12 numaralı kişi: PQ
13 numaralı kişi: PQ
14 numaralı kişi: PQ
15 numaralı kişi: PQ
16 numaralı kişi: PQ
17 numaralı kişi: PQ
18 numaralı kişi: FIFO
19 numaralı kişi: FIFO
20 numaralı kişi: PQ
21 numaralı kişi: PQ
22 numaralı kişi: PQ
23 numaralı kişi: PQ
24 numaralı kişi: FIFO
25 numaralı kişi: FIFO
26 numaralı kişi: FIFO
27 numaralı kişi: PQ
28 numaralı kişi: PQ
29 numaralı kişi: FIFO
30 numaralı kişi: FIFO
31 numaralı kişi: FIFO
32 numaralı kişi: FIFO
33 numaralı kişi: FIFO
34 numaralı kişi: PQ
35 numaralı kişi: FIFO
36 numaralı kişi: FIFO
37 numaralı kişi: FIFO
38 numaralı kişi: FIFO
39 numaralı kişi: PQ
40 numaralı kişi: FIFO
41 numaralı kişi: PQ
42 numaralı kişi: FIFO
Minimum OITS: 528.36

Sadece FIFO asansörü kullanma durumuna göre süre kazancı: 1,098.29
Sadece PQ asansörü kullanma durumuna göre süre kazancı: 5,060.02

f) AsansörPQ'nun hızını iki katına çıkararak ve otelim merdivenini de müşterilere açarak e seçeneğindeki işlemlerin tekrarlanması

EN UYGUN ASANSÖRLER
0 numaralı kişi: Merdiven
1 numaralı kişi: PQ
2 numaralı kişi: Merdiven
3 numaralı kişi: Merdiven
4 numaralı kişi: PQ
5 numaralı kişi: Merdiven
6 numaralı kişi: Merdiven
7 numaralı kişi: Merdiven
8 numaralı kişi: PQ
9 numaralı kişi: FIFO
10 numaralı kişi: FIFO
11 numaralı kişi: Merdiven
12 numaralı kişi: PQ
13 numaralı kişi: Merdiven
14 numaralı kişi: Merdiven
15 numaralı kişi: FIFO
16 numaralı kişi: PQ
17 numaralı kişi: Merdiven
18 numaralı kişi: PQ
19 numaralı kişi: Merdiven
20 numaralı kişi: PQ
21 numaralı kişi: PQ
22 numaralı kişi: FIFO
23 numaralı kişi: Merdiven
24 numaralı kişi: Merdiven
25 numaralı kişi: PQ
26 numaralı kişi: PQ
27 numaralı kişi: PQ
28 numaralı kişi: FIFO
29 numaralı kişi: Merdiven
30 numaralı kişi: PQ
31 numaralı kişi: Merdiven
32 numaralı kişi: PQ
33 numaralı kişi: Merdiven
34 numaralı kişi: Merdiven
35 numaralı kişi: Merdiven
36 numaralı kişi: FIFO
37 numaralı kişi: PQ
38 numaralı kişi: PQ
39 numaralı kişi: PQ
40 numaralı kişi: Merdiven
41 numaralı kişi: FIFO
42 numaralı kişi: FIFO
Minimum OITS: 103.68
Sadece FIFO asansörü kullanma durumuna göre süre kazancı: 1,522.97
Sadece PQ asansörü kullanma durumuna göre süre kazancı: 5,484.70

1.5 Yazılım geliştirme için harcanan süreler

- Tasarım: 3 gün
- Gerçekleştirim: 5 gün
- Test: 2 gün