

Dokumentacja

Dokumentacja projektu „Urząd spraw czworakich” realizowanego w ramach przedmiotu Analiza Algorytmów(AAL).

Opis problemu:

Urząd spraw czworakich przyjmuje petentów w czterech różnych sprawach. Obsługa każdego petenta zajmuje 5 minut i jest wykonywana przez urzędnika specjalizującego się w danej sprawie. Urząd zatrudnia sześciu urzędników, każdy specjalizuje się w dwóch różnych sprawach. W urzędzie znajduje się jedno okienko, przy którym na początku dnia siedzi urzędnik specjalizujący się w pierwszej i drugiej sprawie. Do okienka ustawia się kolejka osób z różnymi sprawami. Gdy do okienka podchodzi osoba ze sprawą inną niż ta, w której specjalizuje się urzędnik, może zostać obsłużona na dwa sposoby: 1) urzędnik przy okienku dzwoni do urzędnika specjalizującego się w tej sprawie i załatwia tę konkretną sprawę telefonicznie, przez co obsługa wydłuża się o dodatkowe 5 minut, lub 2) urzędnik wychodzi i woła na swoje miejsce (na trwałe) urzędnika specjalizującego się w tej sprawie, co także zajmuje dodatkowe 5 minut. Należy znaleźć sposób obsługi petentów minimalizujący całkowity czas obsługi. Porządek ludzi w kolejce jest znany od początku i się nie zmienia.

Skrócony opis problemu:

W kolejce są zadania 4 różnych typów. W urzędzie jest 6 pracowników specjalizujących się w dwóch różnych zadaniach. Jeżeli przy okienku jest urzędnik, który specjalizuje się w danym zadaniu to wykonuje zadanie w 5 minut. Jeśli nie to może wykonać je w 10 minut, lub zawołać kogoś innego na swoje miejsce – trwa to 10 minut (razem z obsługą aktualnego zadania przez zawołanego pracownika). Należy uszeregować pracowników tak, by czas obsługi był jak najmniejszy.

Metody rozwiązywania zadanego problemu:

Metoda Brutalna – wykorzystywana głównie do testowania następnej metody :

Metoda ta sprawdza czy dane zadanie może być wykonane przez aktualnie pracującego pracownika. Jeśli tak to pracownik wykonuje to zadanie, a czas wykonania zadań zwiększa się o 5. W przeciwnym wypadku uruchamiamy cztery razy rekurencyjnie tą samą metodę tylko z innymi zadaniami optymalnie wykonywanymi przechodząc już do następnego zadania. Sprawdzamy jaki czas wykonania jest najmniejszy i takie uszeregowanie pracowników jest optymalne. Czas zwiększa się o 10 (ponieważ albo zawołaliśmy innego pracownika lub wykonaliśmy to zadanie samodzielnie).

Metoda Liniowa – wykorzystywana jako główny algorytm do szeregowania pracowników i obliczania optymalnego czasu obsługi.

Metoda ta najpierw sprawdza czy aktualne zadanie jest obsługiwane optymalnie. Jeśli tak to zwiększa ona czas o 5, nie zmienia pracowników i przechodzi do następnego zadania. Następnie jesteśmy w przypadku takim, że pierwsze zadanie nie jest optymalne, więc sprawdzamy czy następne zadanie też nie jest optymalne. Jeśli również nie jest, to zmieniamy pracownika przed wykonaniem pierwszego zadania. Zwiększamy czas o 10 i sprawdzamy czy następne zadania są takie same jak pierwsze zadanie by szybciej obliczyć czas potrzebny do obsługi tych zadań. Trzecim krokiem jest sprawdzenie następnych zadań gdy pierwsze zadanie nie jest optymalne, a drugie jest optymalne. Należy sprawdzić tu kilka różnych wariantów zadań w kolejce.

Złożoność $O(n)$

Zrzuty ekranu tabeli porównującej złożoność teoretyczną i prawdziwą :

n	t(n)	q(n)			
1000000	35.000	0.875000	51000000	2021.654	0.991007
2000000	73.600	0.920000	52000000	2067.965	0.994214
3000000	113.160	0.943000	53000000	2166.997	1.022168
4000000	152.616	0.953850	54000000	2228.800	1.031852
5000000	191.862	0.959308	55000000	2227.380	1.012445
6000000	233.886	0.974526	56000000	2242.538	1.001133
7000000	271.389	0.969245	57000000	2273.654	0.997217
8000000	310.939	0.971684	58000000	2319.965	0.999985
9000000	349.294	0.970261	59000000	2343.497	0.993007
10000000	396.029	0.990073	60000000	2382.850	0.992854
11000000	449.403	1.021370	61000000	2423.085	0.993068
12000000	488.840	1.018417	62000000	2518.608	1.015568
13000000	537.284	1.033239	63000000	2549.861	1.011850
14000000	560.228	1.000408	64000000	2606.186	1.018041
15000000	629.823	1.049705	65000000	2634.119	1.013123
16000000	650.082	1.015754	66000000	2670.612	1.011595
17000000	679.308	0.998983	67000000	2720.961	1.015284
18000000	709.931	0.986015	68000000	2757.996	1.013969
19000000	746.293	0.981965	69000000	2795.500	1.012862
20000000	804.429	1.005537	70000000	2843.450	1.015518
21000000	877.343	1.044456	71000000	2861.245	1.007481
22000000	894.534	1.016516	72000000	2910.824	1.010703
23000000	940.453	1.022232	73000000	2939.682	1.006741
24000000	984.345	1.025360	74000000	2983.568	1.007962
25000000	993.535	0.993535	75000000	3059.257	1.019752
26000000	1025.953	0.986494	76000000	3082.326	1.013923
27000000	1075.795	0.996107	77000000	3117.133	1.012056
28000000	1129.080	1.008107	78000000	3154.613	1.011094
29000000	1161.808	1.001559	79000000	3136.161	0.992456
30000000	1179.781	0.983151	80000000	3177.216	0.992880
31000000	1218.778	0.982886	81000000	3261.322	1.006581
32000000	1263.278	0.986936	82000000	3262.132	0.994552
33000000	1305.428	0.988960	83000000	3306.113	0.995817
34000000	1419.443	1.043708	84000000	3348.911	0.996700
35000000	1448.644	1.034746	85000000	3412.691	1.003733
36000000	1430.564	0.993448	86000000	3458.569	1.005398
37000000	1469.356	0.992808	87000000	3498.057	1.005189
38000000	1505.236	0.990287	88000000	3516.106	0.998894
39000000	1545.424	0.990656	89000000	3553.111	0.998065
40000000	1587.542	0.992214	90000000	3674.211	1.020614
41000000	1626.054	0.991496	91000000	3681.921	1.011517
42000000	1667.105	0.992325	92000000	3750.192	1.019074
43000000	1708.511	0.993320	93000000	3826.919	1.028742
44000000	1779.151	1.010881	94000000	3785.692	1.006833
45000000	1789.815	0.994342	95000000	3822.069	1.005808
46000000	1850.982	1.005968	96000000	3824.307	0.995913
47000000	1874.498	0.997073	97000000	3867.831	0.996864
48000000	1918.750	0.999349	98000000	4030.083	1.028082
49000000	1942.375	0.991008	99000000	4006.108	1.011644
50000000	1990.537	0.995269			