

알고리즘 설계와 분석 HW1 보고서

(Maximum Subarray Sum)

서강대학교 컴퓨터공학과

20151580

이문형

1. 구현 내용

1) Algorithm 3 ($O(n^4)$: sumtable 을 이용한 알고리즘)

Sumtable[i][j] : 배열의 [0][0], [i][0], [0][j], [i][j] 을 꼭짓점으로 하는 사각형 내의 원소의 합을 의미한다.

이를 이용하여, subrectangle 의 원소들의 합 (이하 sum)을 구할 수 있다.

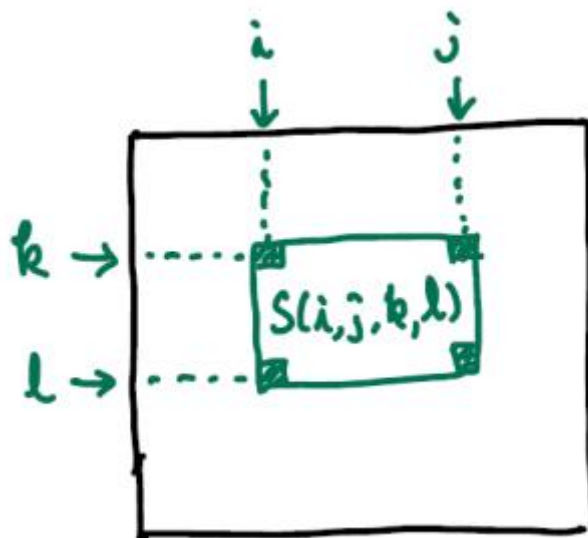


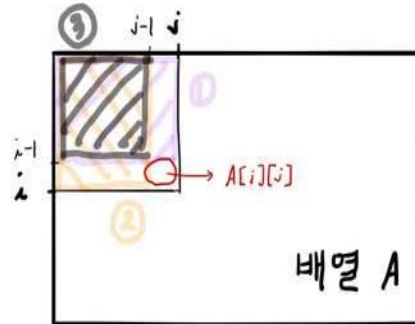
그림 1

위 그림의 $\text{sum}(i, j, k, l)$ 은

$\text{sumtable}[l][j] - \text{sumtable}[l][i-1] - \text{sumtable}[k-1][j] + \text{sumtable}[k-1][i-1]$ 이다.

예외적으로, $i=0$ 이거나 $j=0$ 인 상황에는 배열의 비정상 접근이 발생하므로 sumtable 의 범위를 넘어가면 0 으로 처리해주어 $\text{sum}(i,j,k,l)$ 값을 계산하였다.

Sumtable 을 구성할때도 dynamic 한 방식을 택하였다.



$$\text{sumtable}[i][j] = \begin{cases} \text{sumtable}[i-1][j] + \text{sumtable}[i][j-1] - \text{sumtable}[i-1][j-1] + A[i][j] & (0 < i, j \leq n) \\ \text{sumtable}[i-1][0] + A[i][0] & (j=0) \\ \text{sumtable}[0][j-1] + A[0][j] & (i=0) \\ A[0][0] & (i=0, j=0) \end{cases}$$

그림 2

이 방법을 통하여 가능한 순서쌍 (i,j,k,l) 총 n^4 개의 경우에 sum 값을 비교해서 maximum subarray sum 을 구한다.

2) Algorithm 4 ($O(n^3 \log n)$: 분할정복 방식 적용)

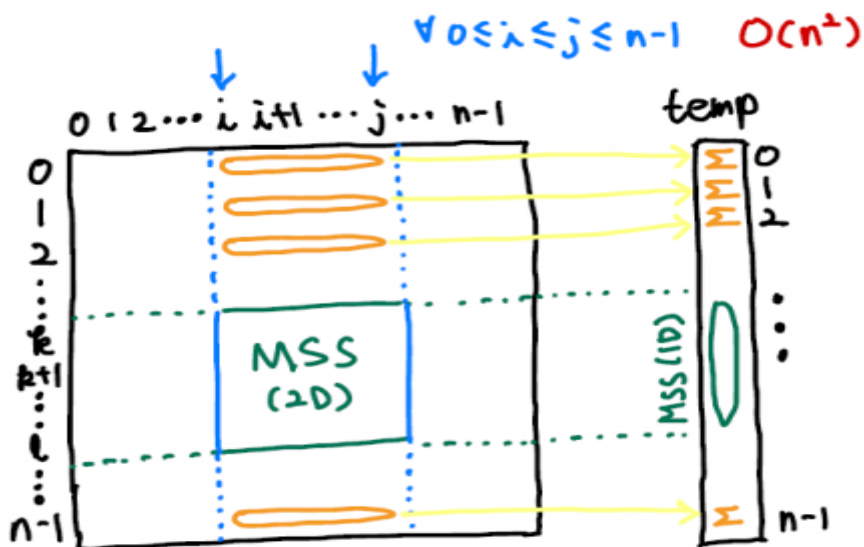


그림 3

그림과 같이 가능한 (i, j) (총 n^2 개)에 대해 temp 열 ($temp[t] = \sum_{k=i}^j A[t][k]$)을 계산하여 temp에 대해 Maximum subsequence를 구한다. 이 방식을 분할 정복을 사용하여 구하면 $O(n \log n)$ 이고 총 n^2 개 경우에 대해 구하므로 $O(n^3 \log n)$ 시간에 배열 A의 maximum subsequence array를 구할 수 있다.

3) Algorithm 5 ($O(n^3)$: kadane 알고리즘 적용)

Algorithm 4번 방식에서 temp 열에 대해 MSS를 구할 때, kadane 알고리즘을 적용하는 방식이다. 그러면 $O(n)$ 시간에 일차원 temp 열에 대해 MSS를 구할 수 있게 되고, 총 $O(n^3)$ 시간에 배열 A의 maximum subsequence array를 구할 수 있다.

2. 실험 환경

OS : Microsoft Windows 10 Home

CPU : Intel(R) Core(TM) i5-10400 CPU @ 2.90GHz, 2901Mhz, 6 코어, 12 논리 프로세서

RAM : 16.0 GB

Compiler : 비주얼 스튜디오 19, release mode / x64 platform

3. 실험 방법

실험 결과의 정확성을 위하여 $N = 10, 100, 200, 300, 400, 500, 800, 1000, 1600, 3200$ 의 $N \times N$ 행렬을 각각 5개씩 생성하였다. 그리고 각각의 데이터에 대해 5번씩 실행을 한 후, 이에 대한 평균을 내어 알고리즘 별, 데이터 별 수행 시간을 구하였다. 또한, 이전 데이터 대비 수행 시간 증가율을 이론적으로 계산한 값과 실제를 비교하여 정확성을 높였다.

수행 시간은 다음과 같은 코드를 통해 계산하였다.

```
#define CHECK_TIME_START QueryPerformanceFrequency ((LARGE_INTEGER*)&freq);  
QueryPerformanceCounter((LARGE_INTEGER*)&start)
```

```
#define CHECK_TIME_END(a) QueryPerformanceCounter(&_LARGE_INTEGER*)&end); a=(float)((float)
(end - start)/freq)
```

```
case 4:

CHECK_TIME_START;
result_2D = solve_4(N, arr);
CHECK_TIME_END(resulttime);

printf("MSS is %d index: %d %d %d %d\n"
```

그림 4

그림 4 는 코드에 수행 시간을 계산하기 위해 삽입된 부분을 나타낸다.

4. 실험 결과

1) 정확도

실험 결과의 정확성을 높이기 위해 각 N 에 대해 다섯개의 데이터를 생성하였고, 각각의 데이터에 대해 모두 5 번씩 수행시간을 측정하였다. 이후 이론적인 시간 증가율 계산값과 실제 수행 시간 증가율을 비교하였다.

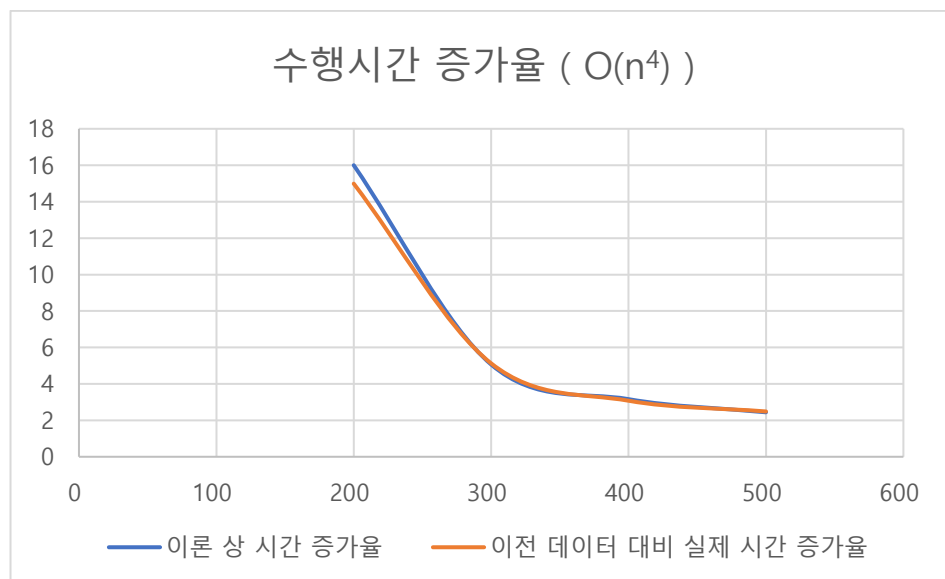


그림 5 알고리즘 3 번 수행시간 증가율 비교

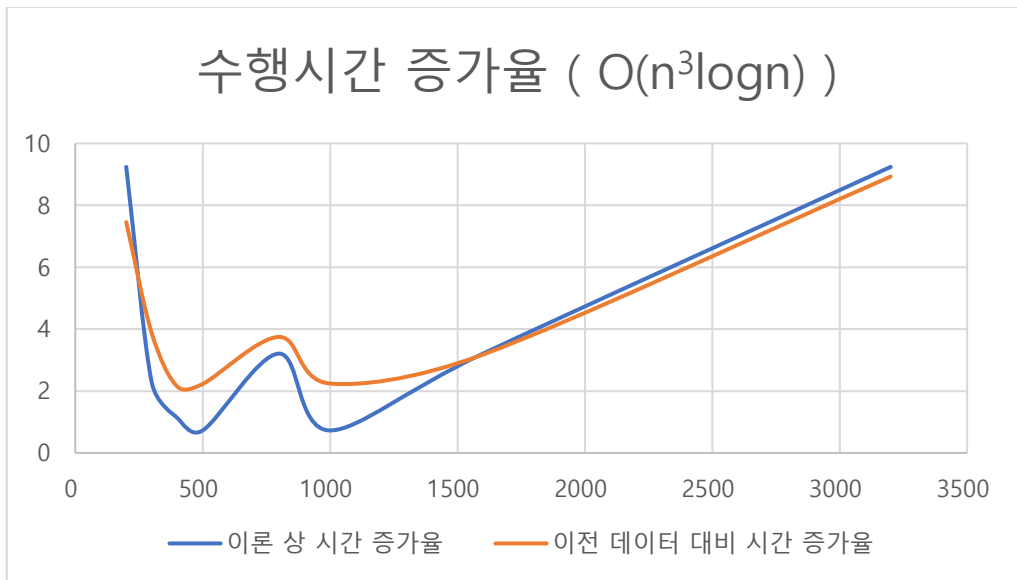


그림 6

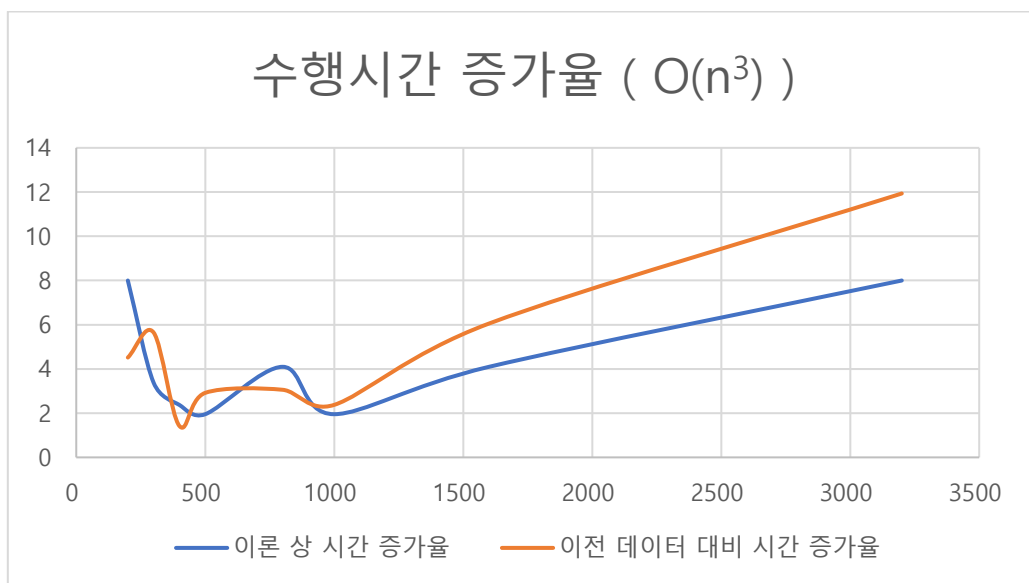


그림 7

3 번, 4 번, 5 번 모두 이론적으로 계산한 값과 비교하여 그래프로 나타냈을 때 모두 비슷한 모습으로 나타났다. (4 번 알고리즘의 이론 상 시간 증가율을 계산할 때, 로그의 base 는 $e^{0.6}$ 을 사용하였다.)

2) 결과

N 이 매우 작은 경우($N = 10$), 오히려 알고리즘 3($O(n^4)$)이 알고리즘 4 보다 더 빠르게 수행하였다. 하지만, N 이 커짐에 따라 알고리즘 3, 알고리즘 4, 알고리즘 5 이 같은 데이터에 대해 프로그램을 수행하는 시간 격차가 커짐을 확인할 수 있었다. (정확한 측정 값은 5 번 항목 참고)

$$O(n^4) > O(n^3 \log n) > O(n^3)$$

N 이 500 까지는 알고리즘 3 에 대해 실행이 가능하였지만 그 이후부터는 합리적인 수행시간을 내놓지 못하였다.

(단위:ms)	$O(n^4)$ 알고리즘	$O(n^3 \log n)$ 알고리즘	$O(n^3)$ 알고리즘
10	0.01072	0.01288	0.00428
100	34.70472	8.80636	1.48276
200	520.22364	65.6492	6.69652
300	2666.02908	254.465	37.79648
400	8200.64548	546.57352	53.59456
500	20391.78236	1218.20804	156.45288
800		4565.3172	478.41668
1000		10234.2314	1136.95216
1600		32528.72892	6883.3378
3200		290457.9798	82156.75292

그림 8

N - t (단위 : ms)의 그래프를 그려보면 다음과 같았다.

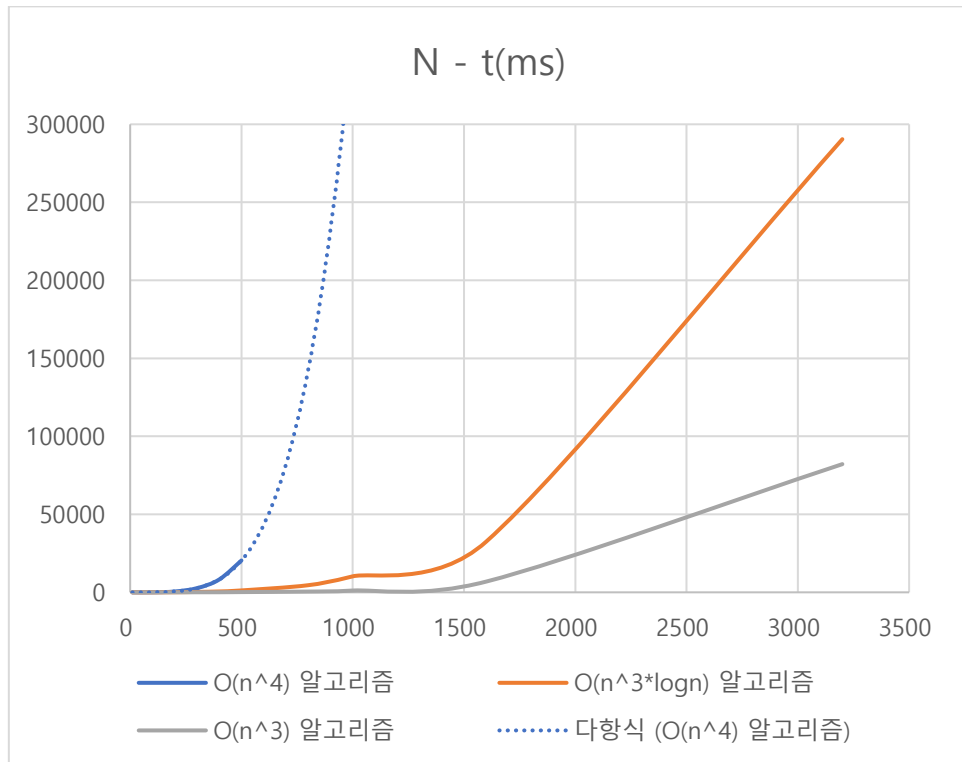


그림 9

3 번 알고리즘 같은 경우에는 N = 500 이후의 데이터를 측정하지 못하여 위와 같이 차수가 4 인 다항식 그래프를 추세선을 이용하여 나타냈다. 알고리즘 3 번과 같은 경우, N 이 커짐에 따라 수행 시간이 급격하게

증가하는 것을 확인할 수 있었다. 알고리즘 4 번과 5 번의 차이는 크게는 10 배 작게는 3.5 배 이상 차이가 났으며 유의미한 격차를 나타냈다.

N의 증가율에 따른 수행시간 증가율을 그래프로 나타내면 다음과 같았다.

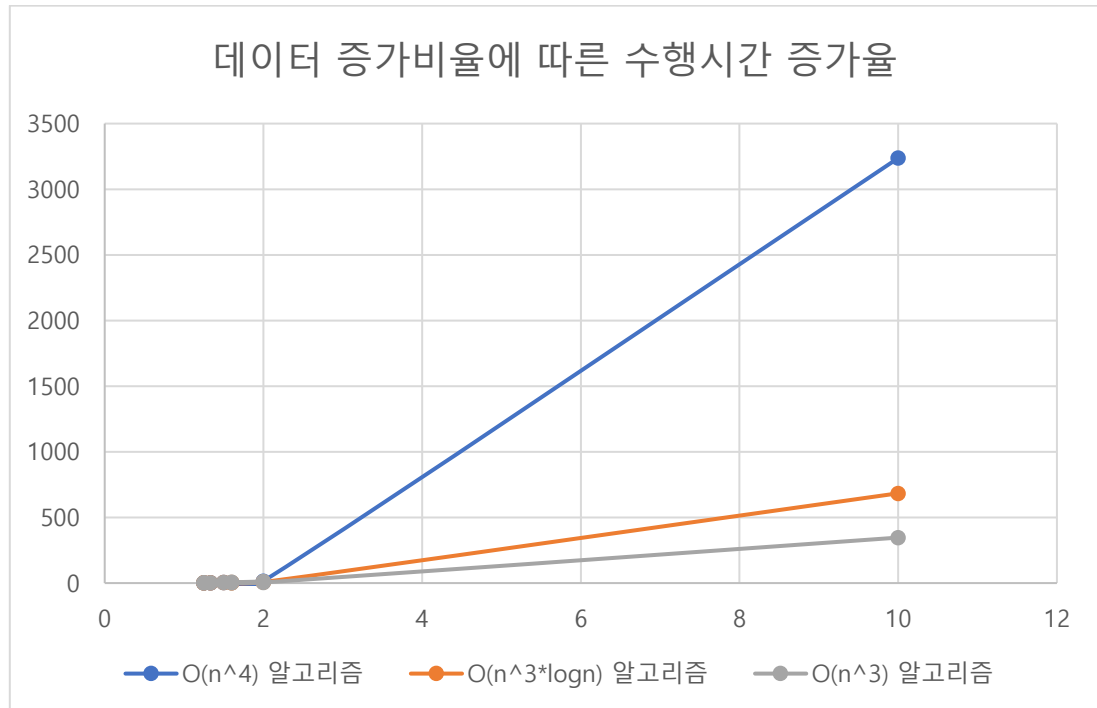


그림 10

N이 증가하는 만큼 모두 수행시간은 증가하였지만, 증가폭은 알고리즘 3 번의 경우가 가장 크게 나타났다.

실제로 n^4 , $n^3 \log n$, n^3 은 그림 8 과 같은 그래프를 나타낸다.

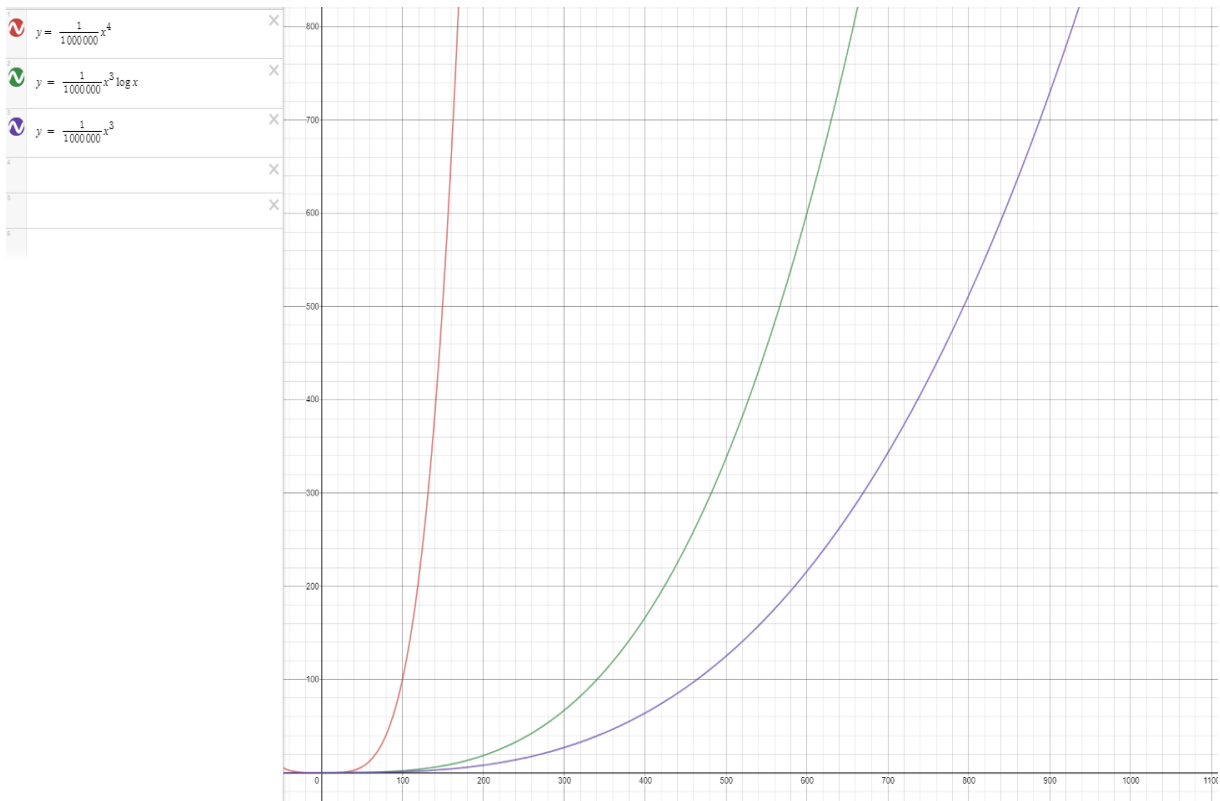


그림 11 (빨간색 : x^4 , 초록색 : $x^3 \log x$, 파란색 : x^3)

그림 8의 그래프와 그림 6 그래프(:실험 결과)가 유사하게 나타났으며, 각 알고리즘의 수행시간과 N의 연관성을 확인할 수 있다.

5. 참고

1) 실험 결과

N	solve_#	1	2	3	4	5	average (ms)
10	3	0.000019	0.000008	0.00001	0.000008	0.000008	0.0106
10	3	0.000017	0.000013	0.000009	0.000009	0.000008	0.0112
10	3	0.000017	0.000008	0.000008	0.000012	0.000007	0.0104
10	3	0.000017	0.000008	0.000009	0.000008	0.000008	0.01
10	3	0.000025	0.000008	0.000008	0.000008	0.000008	0.0114
100	3	0.037433	0.033818	0.034281	0.034047	0.03506	34.9278
100	3	0.035285	0.033469	0.034371	0.035072	0.036017	34.8428
100	3	0.034644	0.033932	0.034074	0.03385	0.034277	34.1554
100	3	0.039722	0.033683	0.033362	0.034458	0.033644	34.9738
100	3	0.035585	0.034104	0.034922	0.034518	0.03399	34.6238
200	3	0.520519	0.515798	0.515138	0.52249	0.517017	518.1924
200	3	0.52249	0.529721	0.51816	0.520411	0.536652	525.4868
200	3	0.518372	0.520411	0.517702	0.529721	0.518291	520.8994
200	3	0.517702	0.518527	0.520519	0.515138	0.51913	518.2032

200	3	0.515138	0.51816	0.52249	0.518372	0.517522	518.3364
300	3	2.744936	2.68558	2.644501	2.712855	2.679329	2693.44
300	3	2.684106	2.645463	2.640629	2.662148	2.677299	2661.929
300	3	2.682559	2.625842	2.65246	2.684556	2.666508	2662.385
300	3	2.686272	2.660602	2.650257	2.661122	2.682066	2668.064
300	3	2.643608	2.618613	2.636361	2.677668	2.645387	2644.327
400	3	8.214275	8.233907	8.238305	8.210896	8.204458	8220.368
400	3	8.157896	8.233775	8.245658	8.16808	8.224375	8205.957
400	3	8.210896	8.173508	8.207519	8.233907	8.170525	8199.271
400	3	8.164988	8.193724	8.196213	8.233907	8.127122	8183.191
400	3	8.14766	8.16808	8.176201	8.245658	8.234604	8194.441
500	3	20.25669	20.34391	20.27693	20.61005	20.4417	20385.86
500	3	20.18318	20.3185	20.33567	20.61012	20.54691	20398.88
500	3	20.30918	20.3332	20.3243	20.5876	20.44058	20398.97
500	3	20.22189	20.29119	20.27895	20.58304	20.42504	20360.02
500	3	20.30992	20.26866	20.27254	20.67876	20.54605	20415.19
10	4	0.000025	0.000016	0.000012	0.000012	0.000016	0.0162
10	4	0.000019	0.000009	0.000009	0.000009	0.000009	0.011
10	4	0.000019	0.000009	0.000011	0.000009	0.000013	0.0122
10	4	0.000019	0.000009	0.000009	0.000009	0.000013	0.0118
10	4	0.000029	0.000009	0.000009	0.00001	0.000009	0.0132
100	4	0.008731	0.008746	0.008713	0.008788	0.008767	8.749
100	4	0.008736	0.008742	0.008763	0.008834	0.008775	8.77
100	4	0.009647	0.008846	0.008874	0.008879	0.008856	9.0204
100	4	0.00883	0.00877	0.00878	0.008832	0.008813	8.805
100	4	0.00875	0.008656	0.008679	0.008673	0.008679	8.6874
200	4	0.066622	0.06636	0.066072	0.064859	0.064906	65.7638
200	4	0.066784	0.065148	0.066329	0.06474	0.065549	65.71
200	4	0.066866	0.06505	0.066514	0.064794	0.06497	65.6388
200	4	0.065888	0.064931	0.068156	0.064875	0.065014	65.7728
200	4	0.065763	0.064984	0.065904	0.064851	0.065301	65.3606
300	4	0.257927	0.254085	0.254241	0.254745	0.254306	255.0608
300	4	0.254767	0.254109	0.254017	0.255459	0.254927	254.6558
300	4	0.254356	0.255168	0.254387	0.25473	0.256625	255.0532
300	4	0.253102	0.253861	0.253978	0.253426	0.253774	253.6282
300	4	0.254023	0.253941	0.2541	0.254304	0.253267	253.927
400	4	0.555119	0.540071	0.55679	0.538842	0.543014	546.7672
400	4	0.553758	0.543587	0.551924	0.54016	0.550852	548.0562
400	4	0.55288	0.541694	0.551923	0.54097	0.556	548.6934
400	4	0.553884	0.539046	0.550678	0.540667	0.541517	545.1584
400	4	0.550473	0.536346	0.554165	0.538816	0.541162	544.1924
500	4	1.215087	1.21236	1.21323	1.215093	1.219559	1215.066

500	4	1.216838	1.212225	1.217681	1.223519	1.217617	1217.576
500	4	1.218008	1.21587	1.216082	1.222246	1.226769	1219.795
500	4	1.214851	1.214613	1.216368	1.221962	1.216544	1216.868
500	4	1.219923	1.218617	1.222722	1.223729	1.223688	1221.736
800	4	4.619484	4.526421	4.668356	4.524608	4.575463	4582.866
800	4	4.580853	4.563112	4.547517	4.533455	4.526609	4550.309
800	4	4.628941	4.554265	4.554586	4.540586	4.542524	4564.18
800	4	4.634149	4.546135	4.547457	4.541648	4.527139	4559.306
800	4	4.633141	4.54956	4.563774	4.549194	4.553953	4569.924
1000	4	10.19316	10.19266	10.18342	10.30834	10.28764	10233.04
1000	4	10.18975	10.17327	10.18869	10.28688	10.31721	10231.16
1000	4	10.20569	10.18614	10.19653	10.29364	10.31704	10239.81
1000	4	10.21611	10.18444	10.17415	10.27764	10.3355	10237.57
1000	4	10.20014	10.17302	10.18259	10.27199	10.32017	10229.58
1600	4	41.02778	40.27371	41.10343	40.79828	40.17347	40675.33
1600	4	41.02582	40.17347	41.20674	40.67025	41.10343	40835.94
1600	4	40.90711	40.18185	40.65168	40.66136	40.18185	40516.77
1600	4	40.78145	40.499	40.53408	40.53925	40.19717	40510.19
1600	4	40.89729	40.19717	40.65752	40.43099	41.02582	40641.76
3200	4	363.5067	359.1449	360.6486	365.2541	361.5317	362017.2
3200	4	364.4295	361.5317	360.5755	362.1955	362.6939	362285.2
3200	4	361.7378	362.6992	361.7675	363.3208	364.4295	362791
3200	4	363.7687	362.6939	365.0569	365.4236	360.5755	363503.7
3200	4	369.2059	359.4266	364.3913	364.6707	363.5067	364240.2
10	5	0.000007	0.000004	0.000003	0.000003	0.000003	0.004
10	5	0.000007	0.000003	0.000003	0.000004	0.000003	0.004
10	5	0.000008	0.000004	0.000003	0.000004	0.000004	0.0046
10	5	0.000009	0.000004	0.000004	0.000004	0.000004	0.005
10	5	0.000007	0.000003	0.000003	0.000003	0.000003	0.0038
100	5	0.001541	0.001524	0.001577	0.001532	0.00154	1.5428
100	5	0.00146	0.001464	0.001475	0.001475	0.001462	1.4672
100	5	0.001459	0.001439	0.001443	0.001482	0.001438	1.4522
100	5	0.00147	0.001468	0.001473	0.001483	0.001467	1.4722
100	5	0.001454	0.001441	0.001439	0.00151	0.001553	1.4794
200	5	0.006702	0.006802	0.006774	0.006667	0.006911	6.7712
200	5	0.006608	0.006664	0.006686	0.006786	0.006756	6.7
200	5	0.006667	0.006625	0.006587	0.006802	0.006597	6.6556
200	5	0.006591	0.006673	0.006823	0.006664	0.006627	6.6756
200	5	0.0066	0.006786	0.0067	0.006608	0.006707	6.6802
300	5	0.036179	0.036258	0.035716	0.036146	0.036113	36.0824
300	5	0.036451	0.036274	0.036637	0.036724	0.036456	36.5084
300	5	0.049347	0.035632	0.035155	0.035347	0.04319	39.7342

300	5	0.036072	0.057292	0.036029	0.035619	0.036721	40.3466
300	5	0.036325	0.036271	0.036414	0.036379	0.036165	36.3108
400	5	0.053504	0.053987	0.054086	0.052536	0.054024	53.6274
400	5	0.053323	0.052574	0.052536	0.054367	0.052574	53.0748
400	5	0.053739	0.054333	0.05377	0.053912	0.053837	53.9182
400	5	0.053125	0.054367	0.052887	0.053987	0.053081	53.4894
400	5	0.053912	0.054049	0.054028	0.053323	0.054003	53.863
500	5	0.156121	0.154963	0.15571	0.156375	0.156481	155.93
500	5	0.156758	0.155686	0.155729	0.157317	0.156332	156.3644
500	5	0.155854	0.156034	0.156632	0.156049	0.155509	156.0156
500	5	0.160049	0.159787	0.158871	0.159547	0.159155	159.4818
500	5	0.154164	0.154397	0.154923	0.154172	0.154707	154.4726
800	5	0.476418	0.482781	0.477557	0.477853	0.47612	478.1458
800	5	0.481809	0.481366	0.48862	0.473033	0.479833	480.9322
800	5	0.473033	0.481759	0.476863	0.477557	0.47629	477.1004
800	5	0.476606	0.477853	0.479798	0.48862	0.478099	480.1952
800	5	0.476541	0.475688	0.479106	0.475688	0.471526	475.7098
1000	5	1.135966	1.13094	1.133161	1.153717	1.138573	1138.471
1000	5	1.144139	1.144815	1.14781	1.158199	1.154568	1149.906
1000	5	1.135714	1.133222	1.135901	1.145231	1.147228	1139.459
1000	5	1.11996	1.126344	1.122427	1.149565	1.13503	1130.665
1000	5	1.123163	1.121391	1.120525	1.129128	1.137087	1126.259
1600	5	6.77513	6.713215	7.185437	6.759054	6.918707	6870.309
1600	5	6.705279	6.759054	7.008526	7.008526	6.897963	6875.87
1600	5	6.820245	6.730594	7.031239	6.713215	7.046705	6868.4
1600	5	6.868412	6.739212	7.012358	6.765826	6.915024	6860.166
1600	5	6.985878	6.765826	7.05569	6.820245	7.082085	6941.945
3200	5	81.33762	82.70216	82.25166	81.33762	82.51503	82028.82
3200	5	81.36657	82.47331	81.94843	82.47331	82.41017	82134.36
3200	5	81.40936	83.13728	82.63752	82.30593	82.8436	82466.74
3200	5	81.19907	82.30593	82.21519	81.36657	82.24667	81866.68
3200	5	81.49835	82.46127	82.5959	82.25166	82.62865	82287.17

● 알고리즘 3 번 결과

N	$O(n^4)$ 알고리즘	이론 상 시간 증가율	이전 데이터 대비 시간 증가율
10	0.01072		
100	34.70472	10000	3237.380597
200	520.22364	16	14.98999675
300	2666.02908	5.0625	5.124774953

400	8200.64548	3.160493827	3.075977506
500	20391.78236	2.44140625	2.486606988

● 알고리즘 4 번 결과

N	$O(n^3 \log n)$ 알고리즘	이론 상 시간 증가율	이전 데이터 대비 시간 증가율
10	0.01288		
100	8.80636	3837.641822	683.7236025
200	65.6492	9.241962407	7.454748614
300	254.465	2.280741233	3.876132535
400	546.57352	1.136521768	2.147932014
500	1218.20804	0.726378748	2.228809109
800	4565.3172	3.208558109	3.74756778
1000	10234.2314	0.726378748	2.24173501
1600	32528.72892	3.208558109	3.178424217
3200	290457.9798	9.241962407	8.929275425

● 알고리즘 5 번 결과

N	$O(n^3)$ 알고리즘	이론 상 시간 증가율	이전 데이터 대비 시간 증가율
10	0.00428		
100	1.48276	1000	346.4392523
200	6.69652	8	4.516253473
300	37.79648	3.375	5.644197285
400	53.59456	2.37037037	1.417977547
500	156.45288	1.953125	2.919193291
800	478.41668	4.096	3.057896282
1000	1136.95216	1.953125	2.376489382
1600	6883.3378	4.096	6.054201788
3200	82156.75292	8	11.93559801

2) 참고 자료

그림 1,3 : 알고리즘 설계와 분석 - 2 반 3 주차 (1) 강의자료