Algoritma Analizi 2. Ödev

k-Way Merge Sort Deneysel Analiz

Öğrenci Adı: Mehmet Ali Duran

Öğrenci Numarası: 21011090

Dersin Eğitmeni: Mehmet Amaç Güvensan

Video Linki: https://youtu.be/yydl4hpt2rk

1- Problemin Çözümü:

Verilen problemde, N boyutlu ve unique değerlerden oluşan bir dizinin k-way merge sort kullanarak sıralanması ve ardından bu sıralama işlemi için karmaşıklık analizinin yapılması istenmiştir. İlk olarak, verilen dizi k alt parçaya bölünmekte ve bu bölme işlemi her alt dizi için tekrar çağrılmaktadır. Böylece her dizi, tek bir eleman kalana dek bölünmeye devam eder. Birleştirme kısmında ise bu parçalar, k birimlik dizilerde sıralanarak yeniden birleştirilir. Sıralama aşamasında heap sort kullanılmıştır. Bu işlem, tüm dizi sıralanana kadar devam eder. Tekrar birleştirme sırasında, farklı k birimlik alt dizilerdeki en küçük elemanı bulmak için heap yapısı kullanılmıştır. Her alt dizinin en küçük elemanı—yani sıralı oldukları için ilk indisteki elemanı—heap tree'ye aktarılır ve min-heap tree oluşturmak için heapify işlemi yapılır. Heap tree'den en küçük eleman seçilerek birleşim dizisinin ilk elemanı olarak atanır. Bu işlem, tüm alt parçalar için tekrarlanır ve en sonunda k-way merge sort kullanılarak sıralanmış dizi elde edilir.

2- Karşılaşılan Sorunlar:

Karşılaşılan en kritik sorunlardan birisi rastgele üretilen sayıların eşsiz olmasını sağlamak. Her adımda 0 – K aralığında rastgele bir sayı üretilse ve bu değer diziye yazılsa birbirinin aynısı olan değerlerin diziye gelme ihtimalleri var. Bu durumu önlemek için şöyle bir yol izlendi. 0 – K aralığındaki sayılar ilk olarak bir dizide tutulur. Daha sonra sıra ile her sayı rastgele bir indisteki sayı ile yer değiştirir. Bu işlem dizi boyunca yapılır ve sonuç olarak içinde her sayıdan bir adet bulunan bir dizi elde edilir.

3- Karmaşıklık Analizi:

```
function kWayMergeSort(array, k):
    if length(array) <= 1:
        return array // Tek elemanlı dizi zaten sıralıdır

    // Diziyi k alt parçaya böl
    subarrays = splitIntoKParts(array, k)

    // Her alt parçayı k-way merge sort ile sırala
    for i from 0 to k - 1:
        subarrays[i] = kWayMergeSort(subarrays[i], k)

    // Alt parçaları birleştir ve sıralı diziyi elde et
    return mergeKSortedArrays(subarrays)

function splitIntoKParts(array, k):
    n = length(array)
    partSize = n / k
    subarrays = []

    // Diziyi k alt parçaya ayır
    for i from 0 to k - 1:
```

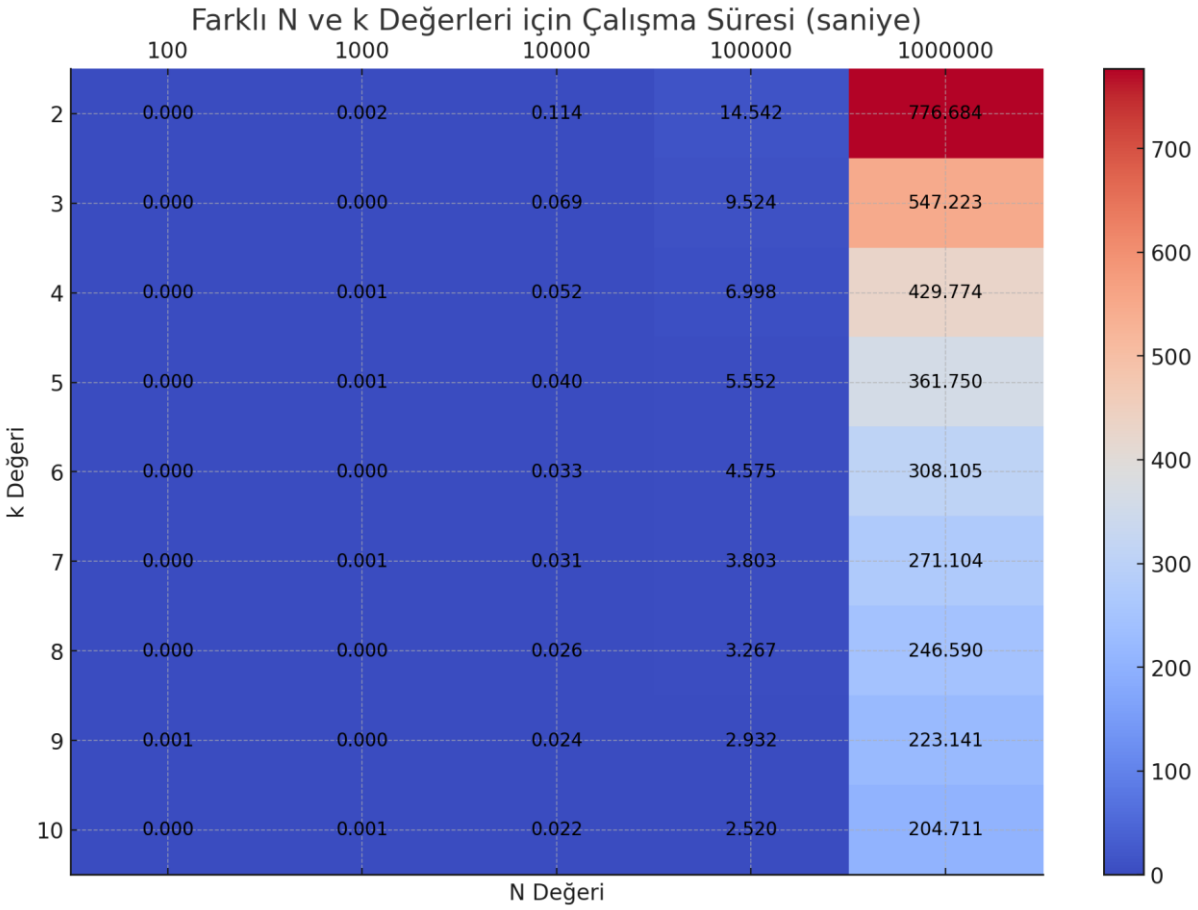
```
        startIndex = i * partSize
        endIndex = startIndex + partSize
        subarrays.append(array[startIndex

])

    return subarrays

function mergeKSortedArrays(subarrays):
    sortedArray = []

    // Her alt diziyi sıralı bir şekilde birleştir
    while any subarray is not empty:
        minElement = findMinimum(subarrays) // Alt dizilerdeki en küçük elemanı bul
        sortedArray.append(minElement)
        remove minElement from its subarray // Alt diziden bu elemanı çıkar

    return sortedArray

function findMinimum(subarrays):
    minElement = infinity

    for each subarray in subarrays:
        if subarray is not empty and subarray[0] < minElement:
            minElement = subarray[0]

    return minElement
```

Kodun karmaşıklık analizini yapmak gerekirse ilk olarak diziyi k alt parçaya bölme işlemi bulunuyor bu işlem logkN maliyet gerektirir. Her alt diziye bölme işleminde diziler kopyalandığından bu da N kadar işlem tutar. Bölme kısmı için maliyet O(N*logkN) olur.

Birleştirme aşamasında her elemanı heap e eklemek ve çıkarmak O(logk) zaman alır. N eleman olduğu için burası için toplam maliyet O(N*logk) olur.

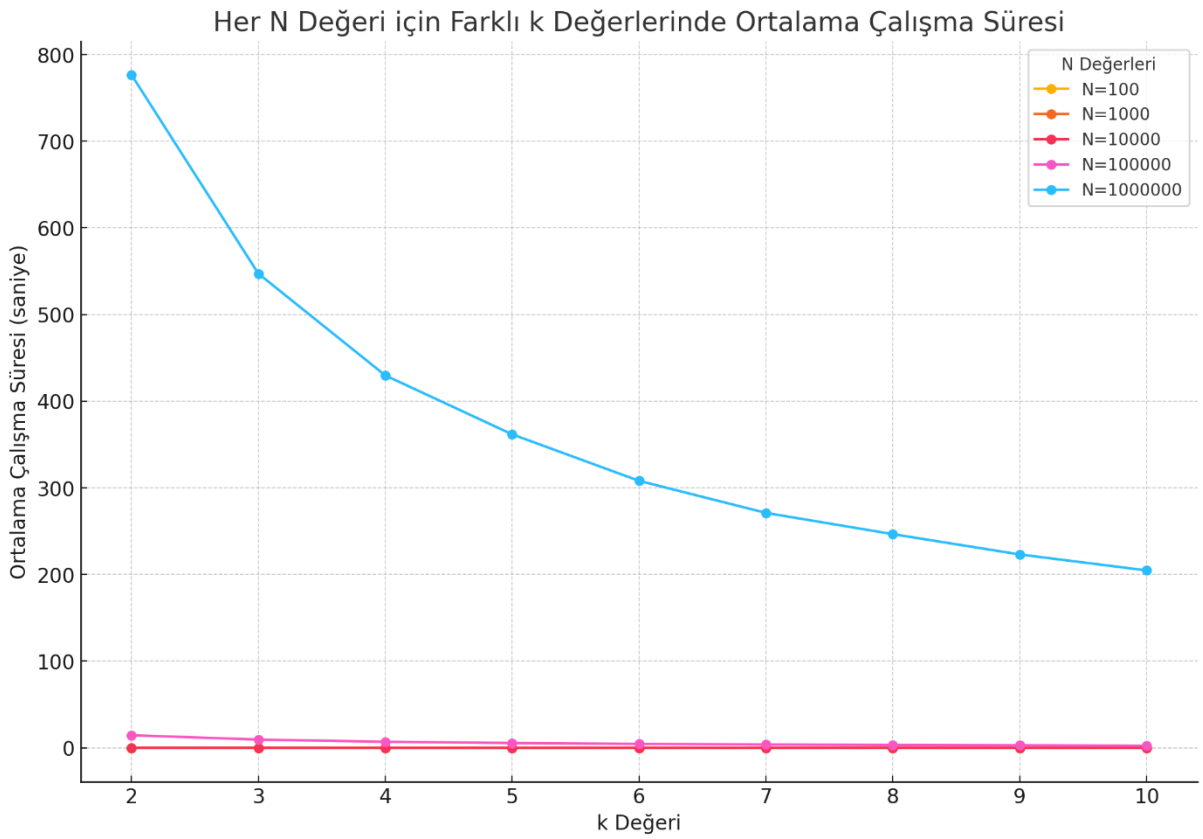
Toplam karmaşıklık ise bu ikisinin toplamından O(N*logkN) + O(N*logk) olur ve bu da O(N*logkN) ya denk gelir.

N ve k değerlerine göre ise sonuçlar şu şekilde olmuştur:

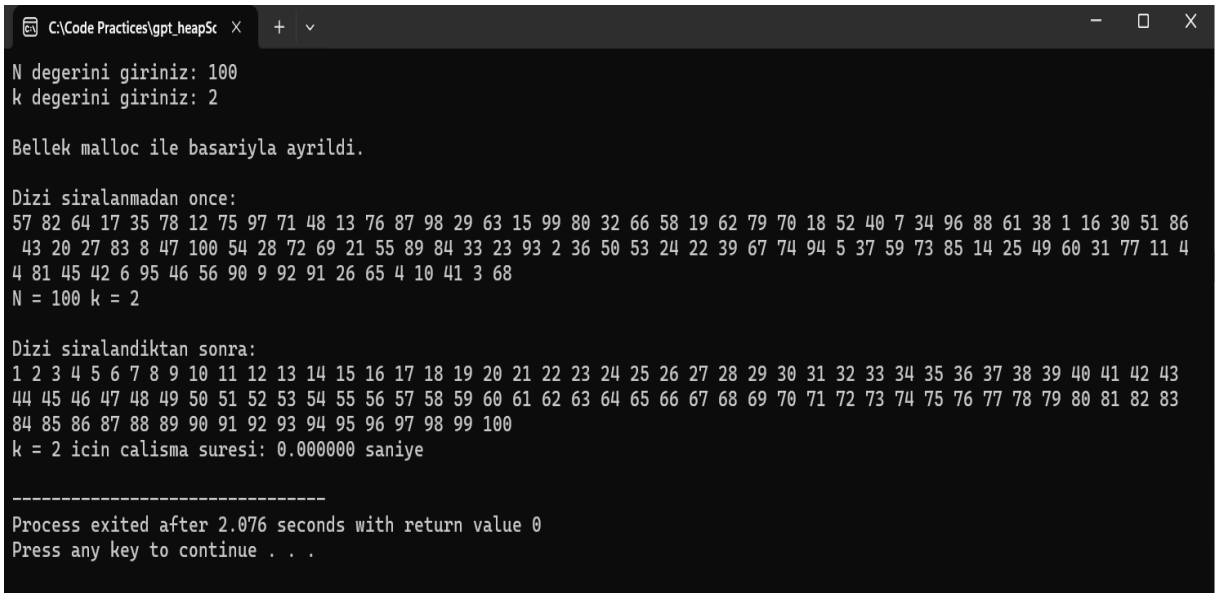


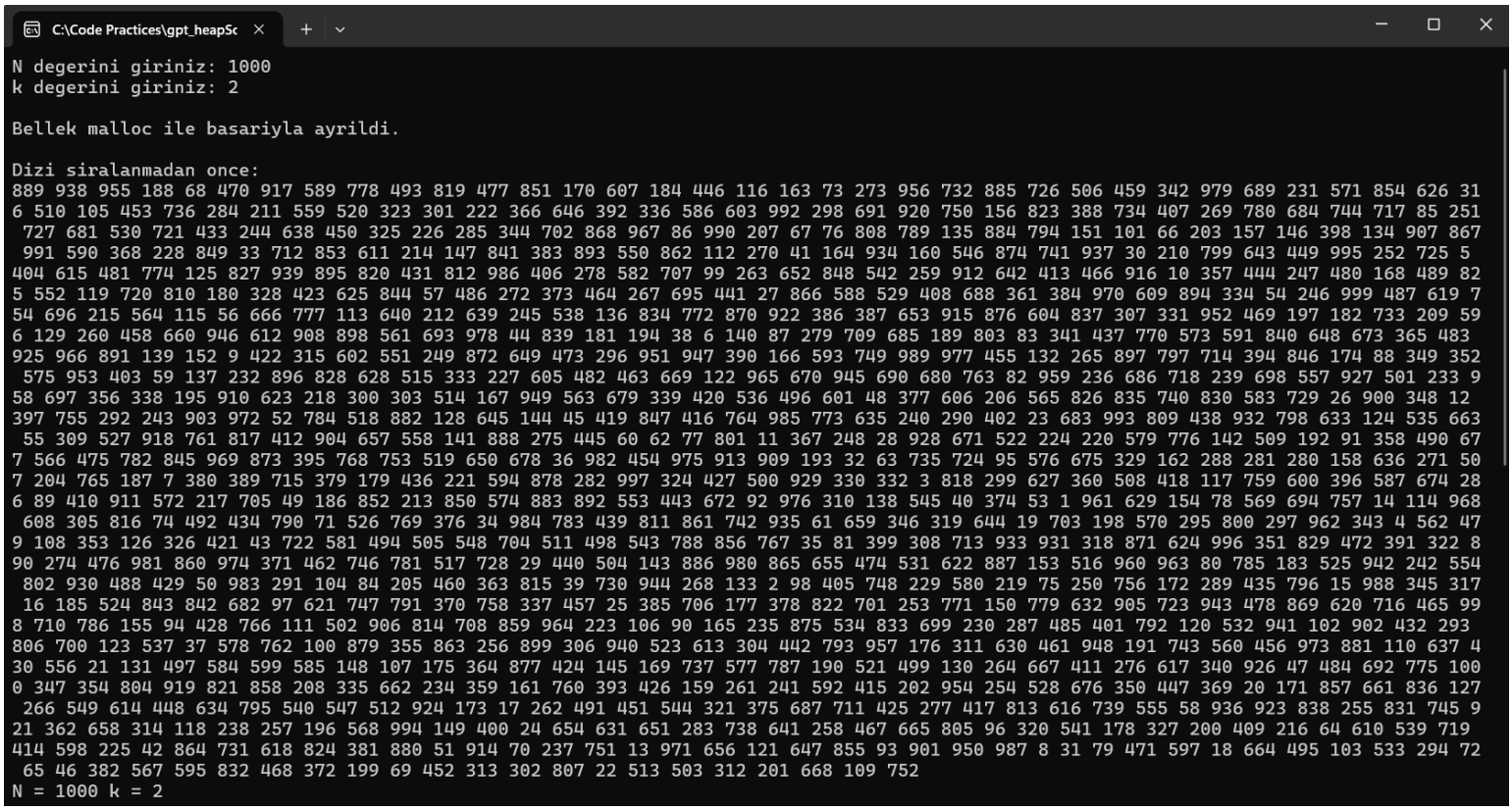
Buradaki ifadeler 10’ar kez çalıştırma için ortalama süreleri ifade etmektedir.

Burada ise k değerlerine göre çalışma sürelerinin nasıl değiştiği gözlemlenmektedir. Görüldüğü gibi k değeri arttıkça çalışma süresi azalmaktadır. Bu da logkN ifadesinden kaynaklanmaktadır. K değeri arttıkça ifadenin değeri küçülür bu da hızlanmasına sebep olur.



4- Ekran çıktıları:





```
Dizi siralandiktan sonra:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 4
9 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94
 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129
130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163
164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197
198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231
232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265
266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299
300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333
334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367
368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401
402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435
436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469
470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503
504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537
538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571
572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605
606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639
640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673
674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707
708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741
742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775
776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809
810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843
844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877
878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911
912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945
946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979
980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000
k = 2 icin calisma suresi: 0.002000 saniye
```

1. Çalıştırma						
		N				
k		100	1000	10000	100000	1000000
	2	0.000000	0.002000	0.114000	14,54200	776,684000
	3	0.000000	0.000000	0.069000	9,524000	547,223000
	4	0.000000	0.001000	0.052000	6,998000	429,774000
	5	0.000000	0.001000	0.040000	5,552000	361,750000
	6	0.000000	0.000000	0.033000	4,575000	308,105000
	7	0.000000	0.001000	0.031000	3,803000	271,104000
	8	0.000000	0.000000	0.026000	3,267000	246,590000
	9	0.001000	0.000000	0.024000	2,932000	223,141000
	10	0.000000	0.001000	0.022000	2,520000	204,711000

2. Çalıştırma						
		N				
k		100	1000	10000	100000	1000000
	2	0.000000	0.001000	0.126000	14,74700	-
	3	0.000000	0.000000	0.067000	9,55100	-
	4	0.000000	0.002000	0.052000	7,39200	-
	5	0.000000	0.001000	0.043000	5,51300	-
	6	0.000000	0.000000	0.032000	4,57200	-
	7	0.000000	0.001000	0.028000	3,88400	-
	8	0.000000	0.000000	0.026000	3,84400	-
	9	0.001000	0.001000	0.024000	2,86000	-
	10	0.000000	0.000000	0.021000	2,51800	-

3. Çalıştırma						
		N				
k		100	1000	10000	100000	1000000
	2	0.000000	0.000000	0.115000	14,66000	-
	3	0.000000	0.000000	0.067000	9,55300	-
	4	0.000000	0.000000	0.050000	7,02200	-
	5	0.000000	0.000000	0.039000	5,64800	-
	6	0.001000	0.000000	0.034000	4,53100	-
	7	0.000000	0.000000	0.028000	3,88500	-
	8	0.000000	0.000000	0.029000	3,26400	-
	9	0.000000	0.000000	0.023000	2,93100	-
	10	0.000000	0.000000	0.022000	2,52100	-

4. Çalıştırma						
	N					
k		100	1000	10000	100000	1000000
	2	0.000000	0.000000	0.112000	15,35600	-
	3	0.000000	0.000000	0.071000	9,47100	-
	4	0.001000	0.000000	0.055000	7,01100	-
	5	0.000000	0.000000	0.039000	5,57300	-
	6	0.000000	0.000000	0.033000	4,53800	-
	7	0.000000	0.000000	0.029000	3,85200	-
	8	0.000000	0.000000	0.027000	3,34900	-
	9	0.000000	0.000000	0.025000	2,87800	-
	10	0.000000	0.000000	0.021000	2,51900	-

5. Çalıştırma						
	N					
k		100	1000	10000	100000	1000000
	2	0.001000	0.001000	0.110000	15,20100	-
	3	0.000000	0.001000	0.080000	10,88500	-
	4	0.000000	0.000000	0.051000	10,54900	-
	5	0.000000	0.001000	0.041000	8,25400	-
	6	0.000000	0.001000	0.032000	6,20800	-
	7	0.000000	0.001000	0.030000	5,32100	-
	8	0.000000	0.000000	0.026000	4,42300	-
	9	0.000000	0.001000	0.023000	2,91100	-
	10	0.000000	0.001000	0.022000	2,49600	-

6. Çalıştırma						
	N					
k		100	1000	10000	100000	1000000
	2	0.001000	0.002000	0.110000	14,479000	-
	3	0.000000	0.001000	0.068000	9,499000	-
	4	0.000000	0.001000	0.050000	7,815000	-
	5	0.000000	0.000000	0.040000	5,529000	-
	6	0.000000	0.001000	0.035000	4,486000	-
	7	0.000000	0.001000	0.028000	3,904000	-
	8	0.000000	0.000000	0.026000	3,874000	-
	9	0.000000	0.001000	0.024000	2,851000	-
	10	0.000000	0.000000	0.022000	2,548000	-

7. Çalıştırma						
		N				
k		100	1000	10000	100000	1000000
	2	0.000000	0.001000	0.121000	14,595000	-
	3	0.000000	0.001000	0.071000	9,483000	-
	4	0.001000	0.001000	0.050000	6,957000	-
	5	0.000000	0.001000	0.039000	5,520000	-
	6	0.000000	0.001000	0.033000	4,443000	-
	7	0.000000	0.001000	0.029000	3,745000	-
	8	0.000000	0.000000	0.025000	3,268000	-
	9	0.000000	0.000000	0.023000	2,900000	-
	10	0.000000	0.001000	0.021000	2,474000	-

8. Çalıştırma						
		N				
k		100	1000	10000	100000	1000000
	2	0.000000	0.001000	0.111000	15,227000	-
	3	0.000000	0.001000	0.069000	9,481000	-
	4	0.000000	0.000000	0.050000	7,075000	-
	5	0.000000	0.001000	0.041000	5,577000	-
	6	0.000000	0.001000	0.033000	4,442000	-
	7	0.002000	0.001000	0.032000	3,820000	-
	8	0.000000	0.000000	0.026000	3,301000	-
	9	0.000000	0.001000	0.025000	2,852000	-
	10	0.000000	0.000000	0.021000	2,592000	-

9. Çalıştırma						
		N				
k		100	1000	10000	100000	1000000
	2	0.000000	0.000000	0.114000	14,58700	-
	3	0.000000	0.000000	0.070000	9,51800	-
	4	0.000000	0.000000	0.054000	6,96600	-
	5	0.000000	0.000000	0.042000	5,42700	-
	6	0.000000	0.000000	0.034000	4,45500	-
	7	0.000000	0.000000	0.030000	3,78400	-
	8	0.000000	0.000000	0.026000	3,54200	-
	9	0.000000	0.000000	0.023000	3,54800	-

	10	0.000000	0.000000	0.022000	2,53700	-

10. Çalıştırma						
	N					
k		100	1000	10000	100000	1000000
	2	0.000000	0.001000	0.110000	14,625000	-
	3	0.000000	0.001000	0.069000	9,424000	-
	4	0.000000	0.000000	0.050000	10,929000	-
	5	0.000000	0.001000	0.044000	8,017000	-
	6	0.000000	0.000000	0.033000	4,548000	-
	7	0.000000	0.001000	0.029000	3,885000	-
	8	0.000000	0.000000	0.025000	3,416000	-
	9	0.000000	0.001000	0.023000	2,806000	-
	10	0.000000	0.000000	0.022000	2,608000	-