

Homework Assignment

Algorithm Design

Bujor David Marian

CEN 1.1A

Faculty of Automatics, Computer Science and Electronics

May 2024

<https://github.com/BujorDavidMarian/Homework-Assignment>

1) Problem Statement

Imagine a scenario where three employees are tasked with combing through a shelf full of books to find specific information. To ensure the work is divided both fairly and efficiently, the books need to be allocated among the three without the hassle of moving them around or creating separate piles. The simplest solution is to split the shelf into three distinct sections, with each section assigned to one employee. But how can we ensure the division is equitable?

If every book were the same size, the solution would be straightforward: divide the shelf into equal sections, like so: 100 100 100 – 100 100 100 – 100 100 100

This way, each person faces 300 pages. However, the challenge arises when books vary in size. Using the same method for a shelf with books of the following sizes: 100 200 300 – 400 500 600 – 700 800 900

It's clear that the distribution is uneven. The first section has a total of 600 pages, whereas the last one contains a whopping 2,400 pages. In such cases, I'd definitely prefer the first section!

To achieve a more balanced workload, a better division might look something like this: 100 200 300 400 500 – 600 700 – 800 900

In this arrangement, the heaviest load a person would have is 1,700 pages, and the lightest is 1,300 pages. This example illustrates the need for a strategic approach to distribute workloads as evenly as possible when dealing with tasks of varying difficulty or size.

The goal is to determine an optimal division of books such that the maximum number of pages assigned to any single employee is minimized.

2) Tehnical Report

2.1) Algorithm

2.2) Computation Time

2.3) Explanation

Pseudocode for the Algorithms

Algorithm for Checking the Feasibility of a Division

This algorithm checks if it is possible to divide the books among employees such that no employee has a workload greater than a given value 'mid'.

Function 'isFeasible':

IS-FEASIBLE(books, n, k, mid)

1. requiredEmployees = 1 (O(1))
2. currentLoad = 0 (O(1))
3. for i = 0 to n - 1 do (O(n))
4. if books[i] > mid then (O(1))
5. return 0 (O(1))
6. if currentLoad + books[i] > mid then (O(1))
7. requiredEmployees = requiredEmployees + 1 (O(1))
8. currentLoad = books[i] (O(1))
9. if requiredEmployees > k then (O(1))
10. return 0 (O(1))

11. else

($O(1)$)

12. currentLoad = currentLoad + books[i] ($O(1)$)

13. return 1

($O(1)$)

Complexity Analysis:

Lines 1 and 2 have complexity $O(1)$.

The 'for' loop (lines 3-13) iterates through all books once, so it has a complexity of $O(n)$.

Lines 4, 6, 7, 8, 9, 11, and 12 each have a complexity of $O(1)$.

The total complexity for 'isFeasible' is $O(n)$.

Algorithm for Finding the Optimal Workload

This algorithm uses binary search to find the smallest maximum workload that any employee has to bear.

Function 'findOptimalWorkload':

FIND-OPTIMAL-WORKLOAD(books, n, k)

1. sum = 0	(O(1))
2. for i = 0 to n - 1 do	(O(n))
3. sum = sum + books[i]	(O(1))
4. left = 0	(O(1))
5. right = sum	(O(1))
6. result = INT_MAX	(O(1))
7. while left <= right do (O(log(sum)))	
8. mid = (left + right) / 2	(O(1))
9. if IS-FEASIBLE(books, n, k, mid) then	(O(n))
10. result = mid	(O(1))
11. right = mid - 1	(O(1))
12. else	(O(1))
13. left = mid + 1	(O(1))
14. return result	(O(1))

Complexity Analysis:

Lines 1 and 4-6 have complexity $O(1)$.

The 'for' loop (lines 2-3) iterates through all books once, so it has a complexity of $O(n)$.

The 'while' loop (lines 7-13) performs a binary search over the range $[0, \text{sum}]$.

The binary search performs $O(\log(\text{sum}))$ iterations.

Each iteration calls 'isFeasible', which has a complexity of $O(n)$.

The total complexity for 'findOptimalWorkload' is $O(n \log(\text{sum}))$.

Algorithm for Dividing the Books Among Employees

This algorithm uses the optimal value found by 'findOptimalWorkload' to divide the books among employees in a way that minimizes the maximum workload.

Function 'divideBooksAmongEmployees':

DIVIDE-BOOKS-AMONG-EMPLOYEES(books,
num_books, num_employees)

1. optimalWorkload = FIND-OPTIMAL-
WORKLOAD(books, num_books, num_employees)
($O(n \log(\text{sum}))$)
2. PRINT "The optimal division of books to minimize the
maximum workload is:" ($O(1)$)
3. currentLoad = 0 ($O(1)$)
4. currentEmployee = 1 ($O(1)$)
5. startIdx = 0 ($O(1)$)
6. for i = 0 to num_books - 1 do ($O(n)$)
7. if currentLoad + books[i] > optimalWorkload then
 ($O(1)$)
8. PRINT "Employee", currentEmployee, ":",
 currentLoad, "pages (" ($O(1)$)
9. for j = startIdx to i - 1 do ($O(n)$)
10. PRINT books[j] ($O(1)$)
11. PRINT ")" ($O(1)$)
12. currentLoad = books[i] ($O(1)$)
13. currentEmployee = currentEmployee + 1 ($O(1)$)
14. startIdx = i ($O(1)$)
15. else ($O(1)$)
16. currentLoad = currentLoad + books[i] ($O(1)$)

```

17.  if i == num_books - 1 then                                (O(1))
18.      PRINT "Employee", currentEmployee, ":",
currentLoad, "pages ("                                         (O(1))
19.      for j = startIdx to i do                               (O(n))
20.          PRINT books[j]                                     (O(1))
21.      PRINT ")"
(O(1))
22. PRINT "The optimal division of books to minimize the
maximum workload is:", optimalWorkload, "pages" (O(1))

```

Complexity Analysis:

Line 1 calls 'findOptimalWorkload', which has a complexity of $O(n \log(\text{sum}))$.

Lines 2-4 have complexity $O(1)$.

The 'for' loop (lines 5-20) iterates through all books once, so it has a complexity of $O(n)$.

Lines 6-10 and 16-20 contain operations with complexity $O(1)$ each, and other loops that iterate through a subset of the books, thus having a total complexity of $O(n)$.

The total complexity for 'divideBooksAmongEmployees' is $O(n \log(\text{sum}) + n) = O(n \log(\text{sum}))$.

Explanation

- IS-FEASIBLE Algorithm: This algorithm determines if an employee can manage a given sequence of books without their workload exceeding a specified threshold, denoted as 'mid'.
- FIND-OPTIMAL-WORKLOAD Algorithm: This algorithm employs a binary search method to identify the smallest possible maximum workload that an employee must handle. It finds the optimal workload threshold that balances the distribution of books among employees.
- DIVIDE-BOOKS-AMONG-EMPLOYEES Algorithm: This algorithm uses the optimal workload value identified by the previous algorithm to allocate the books among the employees. It ensures that the distribution respects the optimal workload and then outputs the results.

Total Complexity

The primary function responsible for distributing the books is `divideBooksAmongEmployees`. Therefore, the overall complexity of the algorithm is:

$$T(n) = O(n \log(\text{sum}))$$

Here, n represents the number of books, and sum represents the total number of pages in all books.

Summary of Steps:

Complexity Calculation for 'isFeasible':

Individual lines: $O(1)$

'For' loop: $O(n)$

Total: $O(n)$

Complexity Calculation for 'findOptimalWorkload':

Sum of individual lines and 'for' loop: $O(n)$

'While' loop with binary search and calls to 'isFeasible':
 $O(n \log(\text{sum}))$

Total: $O(n \log(\text{sum}))$

Complexity Calculation for
'divideBooksAmongEmployees':

Call to 'findOptimalWorkload': $O(n \log(\text{sum}))$

Sum of individual lines and 'for' loops: $O(n)$

Total: $O(n \log(\text{sum}))$

3) Experimental Data

3.1) Results & Conclusions

3.2) Analyzing the time complexity

Algorithms for Data Generation

Function 'generateAndSaveBooks'

GENERATE-AND-SAVE-BOOKS(books, num_books)

1. srand(time(NULL)) (O(1))
2. a = 100 (O(1))
3. for i = 0 to num_books - 1 do (O(num_books))
4. books[i] = a (O(1))
5. a += 100 (O(1))
6. file = fopen("book_data.txt", "w") (O(1))
7. for i = 0 to num_books - 1 do (O(num_books))
8. PRINT books[i] (O(1))
9. fclose(file) (O(1))

Complexity Analysis:

Lines 1-2 have a complexity of $O(1)$.

The 'for' loop (lines 3-5) iterates over all books once, so it has a complexity of $O(\text{num_books})$.

Line 6 has a complexity of $O(1)$.

The 'for' loop (lines 7-8) iterates over all books once to write data to the file, so it has a complexity of $O(\text{num_books})$.

Line 9 has a complexity of $O(1)$.

The total complexity for 'generateAndSaveBooks' is $O(\text{num_books})$.

Function 'generateAndSaveRandomBooks'

GENERATE-AND-SAVE-RANDOM-BOOKS(*books*,
num_books)

1. `srand(time(NULL))` $(O(1))$
2. `for i = 0 to num_books - 1 do`
 $(O(\text{num_books}))$
3. `books[i] = rand() % 1000000000 + 1` $(O(1))$
4. `file = fopen("random_book_data.txt", "w")` $(O(1))$
5. `for i = 0 to num_books - 1 do` $(O(\text{num_books}))$
6. `PRINT books[i]` $(O(1))$
7. `fclose(file)` $(O(1))$

Complexity Analysis:

Line 1 has a complexity of $O(1)$.

The 'for' loop (lines 2-3) iterates over all books once, so it has a complexity of $O(\text{num_books})$.

Line 4 has a complexity of $O(1)$.

The 'for' loop (lines 5-6) iterates over all books once to write data to the file, so it has a complexity of $O(\text{num_books})$.

Line 7 has a complexity of $O(1)$.

The total complexity for 'generateAndSaveRandomBooks' is $O(\text{num_books})$.

Testing Functions

Function 'testDataGeneration'

TEST-DATA-GENERATION()

1. `num_books = 0` $(O(1))$
2. `books[MAX_BOOKS]` $(O(1))$
3. `PRINT "Enter the number of books for data generation: "`
 $(O(1))$
4. `SCAN num_books` $(O(1))$
5. `GENERATE-AND-SAVE-BOOKS(books, num_books)`
 $(O(\text{num_books}))$

6. PRINT "Book data has been generated and saved to
book_data.txt\n" (O(1))

Complexity Analysis:

Lines 1-4 and 6 have a complexity of O(1).

Line 5 calls 'generateAndSaveBooks', which has a complexity of O(num_books).

The total complexity for 'testDataGeneration' is O(num_books).

Function 'randomtestDataGeneration'

RANDOM-TEST-DATA-GENERATION()

1. num_books = 0 (O(1))
2. books[MAX_BOOKS] (O(1))
3. PRINT "Enter the number of books for random data
generation: " (O(1))
4. SCAN num_books (O(1))
5. GENERATE-AND-SAVE-RANDOM-BOOKS(books,
num_books) (O(num_books))

6. PRINT "Random book data has been generated and saved
to random_book_data.txt\n" (O(1))

Complexity Analysis:

Lines 1-4 and 6 have a complexity of O(1).

Line 5 calls 'generateAndSaveRandomBooks', which has a complexity of O(num_books).

The total complexity for 'randomtestDataGeneration' is O(num_books).

Explanation

GENERATE-AND-SAVE-BOOKS: This function generates an array of books with page numbers multiple of 100 and saves the data to a file.

GENERATE-AND-SAVE-RANDOM-BOOKS: This function generates an array of books with random page numbers between 1 and 100,000,000 and saves the data to a file.

TEST-DATA-GENERATION: This function initializes and collects the number of books from the user, then generates and saves the data.

RANDOM-TEST-DATA-GENERATION: This function initializes and collects the number of books from the user, then generates and saves random data.

Total Complexity

The data generation and testing functions have the following complexities:

'generateAndSaveBooks': $O(\text{num_books})$

'generateAndSaveRandomBooks': $O(\text{num_books})$

'testDataGeneration': $O(\text{num_books})$

'randomtestDataGeneration': $O(\text{num_books})$

Test Number 1:

Input:

3

9

100 150 200 250 300 350 400 450 500

Output:

The optimal division of books to minimize the maximum workload is:

Employee 1: 1000 pages (100 150 200 250 300)

Employee 2: 750 pages (350 400)

Employee 3: 950 pages (450 500)

The optimal division of books to minimize the maximum workload is: 1000 pages

Time: 0.014 s

Test Number 2:

Input:

3

1000

10485 27686 15546 2615 31494 6137 16749 6360 22833 16010 27840 4880 27193 12195 17186 20511 15020 20136 16186 12704 28938
1576 18219 21436 3369 6578 1815 3486 14040 18922 21730 22054 24515 32475 312 24802 15511 27791 14801 1850 22886 26077 24819
20970 19788 20867 32313 14006 20819 30069 11412 24152 16020 3188 10639 1726 14239 23009 27095 21874 285 12244 22266 11866
19892 7017 6767 25249 299 11098 6615 19445 11983 1235 2178 10227 22293 14977 12853 18399 10368 6637 25386 26777 18373 25352
4441 18124 25390 7686 10998 17257 5436 880 11835 11348 23184 10869 28263 32093 7947 29058 10759 11633 9781 21778 22132
23569 21211 10438 17565 29444 4605 26296 774 4169 27879 2317 8465 19536 1693 8649 14205 12394 2172 9447 23084 29053 16927
6314 22313 17148 128 8238 26345 22038 24671 5186 12753 22830 8168 7802 20217 2956 13200 2687 8240 14674 8768 16810 12096
13233 27829 9425 28787 6492 1063 14433 3669 829 5160 22189 8892 26984 9168 23686 262 21581 21015 1036 30114 11825 12674
25027 7890 18655 1794 14314 17070 27546 28697 5128 18029 32087 13374 27335 10976 7977 17795 9733 12863 19197 13429 22825
4904 19079 6979 4019 23781 6780 17443 27160 28910 3586 3826 31948 13742 18021 23066 9632 26250 21259 29369 19278 9504 3498
20480 15186 12236 1720 7641 6652 24945 5667 23244 6340 24182 25667 3363 26333 4878 32525 21324 4109 26301 20361 21376 15803
18712 902 20130 3658 31692 16268 23146 30373 15841 11538 1730 25115 15424 14577 29296 1134 27507 5657 4854 4646 11949 22658
13238 21291 28682 15462 28762 17299 3607 28078 11891 14009 17045 25514 23136 12431 26234 9476 27506 26933 14137 2039 29518
11978 323 17084 15506 6530 21454 24645 15547 17912 8661 27950 23878 2502 29792 28913 24597 22100 4613 1881 25365 15853
16303 11833 18206 20248 32414 7005 32282 14885 27567 11969 1130 29958 9182 21761 2437 20832 21001 28177 3042 14562 18935
3118 27673 982 21656 7956 15173 7669 4494 24434 18604 28032 4589 3739 12962 12428 4159 24190 32426 17566 27084 5592 31967
3191 15640 17576 31378 26673 20908 7396 2101 29579 11545 17258 22135 32386 31035 11577 28337 29729 8123 23777 19073 9113
16087 5176 3764 8084 23398 20047 28010 10183 29718 5540 29556 20678 23033 2713 17665 24093 29248 11813 3450 668 17734 17610

9898 31474 32464 26055 15138 13950 17299 2702 26588 609 32758 12920 8478 27352 5146 28344 24225 18121 279 27278 21235 8603
31625 17341 12826 30539 1868 18238 19322 30396 131 26166 31481 15951 13791 30502 13664 18100 30750 5985 13316 28672 17105
16580 935 25855 25187 26529 31385 18635 15191 13564 23074 25245 5841 12538 7973 15332 6232 29078 17917 25748 16602 8598
1451 20929 8944 29629 17971 2668 11364 22083 22083 5789 10481 31362 9238 25778 25161 1266 32152 6390 349 31063 7115 31605
5336 14137 32105 30735 24812 6805 21707 25804 23634 5330 3337 623 32396 20982 10851 27470 17262 15898 2527 25696 2012 32540
17460 17560 9173 13971 14311 26225 26033 8200 32436 22475 21158 24419 21039 15349 20625 5027 9131 5131 7411 466 27483 24962
13626 27704 29367 22273 5187 27088 7713 2930 26908 20984 31177 25434 11022 4204 3534 9317 11995 30069 15782 26438 8516
22110 14015 23485 2766 4648 12902 4857 16582 20095 3817 17441 26499 236 1016 16174 27115 6219 30289 507 2617 23168 11933
9119 28669 19046 30612 2554 22371 10760 26472 7566 17849 27924 22625 28560 26412 12172 21700 1599 31186 1215 31226 17269
27118 12920 5870 6524 879 11381 2487 546 23737 25626 8311 27664 24232 16487 387 4695 3538 21032 24260 29260 30989 6251
30813 21398 28978 26935 12941 5765 28192 21115 17111 15526 17667 1861 14433 24468 5078 11572 25499 2576 1574 7728 24864
10068 22297 29272 19425 31909 13488 26172 23168 24872 26697 29005 27642 10230 26728 8338 18313 28786 14262 2460 5065 20699
13642 27067 18713 3869 16936 17778 9028 23533 24583 7209 22113 17428 11608 1058 2807 19467 25219 17070 25957 27081 13428
13473 7297 20697 16470 6915 25472 17583 25758 8200 11030 25916 11084 10306 32221 10688 10605 17430 31739 32224 30693 21869
14848 13560 10603 15787 1805 18354 25045 10213 24123 4056 29334 17231 1850 30310 8325 499 10809 18404 30432 12201 23302
8946 19344 30019 24439 13605 22702 7176 13576 14865 3911 9572 6597 28839 1884 2781 7275 31426 12589 14921 30666 23292 11192
8766 15725 1222 25724 25291 4989 12984 25730 30309 2531 30501 28318 26404 19117 15091 21235 25648 10675 7849 7093 19318
22394 29130 30515 31647 15192 13410 788 578 14553 22238 23776 31320 25698 11296 7107 12017 27412 16851 6030 22054 12787
4860 8447 30799 28744 18642 3906 22390 18756 32208 18103 27403 10469 14088 7414 17368 26654 1978 15889 27704 31794 4901
30489 6276 13665 24733 2531 15311 20678 13790 28171 21715 24960 27552 9018 1834 21776 28780 30885 21698 5890 28077 28118
2955 6096 17162 4271 15218 15107 19443 31916 18579 28827 9082 24424 28074 32177 23141 9827 17099 14427 21377 14850 7252
15056 14905 27095 22047 20940 18278 28195 24316 10806 18537 8857 6238 513 7288 24165 24148 24656 7221 3437 19387 6365 8069
28375 3763 7835 5535 7371 24911 27034 25443 16762 31377 29739 31726 7519 2965 4934 26475 17092 7436 11816 11307 10889 24812
6464 19961 1569 19558 6220 22439 29405 3609 19392 8941 29293 13445 3431 21108 19620 6338 76 12845 15995 24945 22491 29159
6745 12788 6724 28193 15761 23002 25120 31516 32556 6301 25431 26165 31722 18832 14145 8562 25475 3091 13323 19803 11058
15369 21341 5387 18804 20108 27705 4230 24334 26335 17447 8525 29005 14243 15247 32048 6545 1102 6502 23396 19981 22234
14860 30677 10184 15802 9589 24765 18919 9746 20616 17014 20015 9767 3524 9613 23238 19477 28969 21271 17721 17600 11094
21320 26307 28428 8175 14913 1485 12113 15169 14122 30897 5897 19241 6529 26051 13516 24369 16561 17554 8319 29693 10727
1131 6088 28134 4129 31775 16068 9956 21128 1798 22781 5041 5375 29508 3734

Output:

The optimal division of books to minimize the maximum workload is:

Employee 1: 5517714 pages (10485 27686 15546 2615 31494 6137 16749 6360 22833 16010 27840
4880 27193 12195 17186 20511 15020 20136 16186 12704 28938 1576 18219 21436 3369 6578 1815 3486 14040 18922 21730 22054
24515 32475 312 24802 15511 27791 14801 1850 22886 26077 24819 20970 19788 20867 32313 14006 20819 30069 11412 24152 16020
3188 10639 1726 14239 23009 27095 21874 285 12244 22266 11866 19892 7017 6767 25249 299 11098 6615 19445 11983 1235 2178
10227 22293 14977 12853 18399 10368 6637 25386 26777 18373 25352 4441 18124 25390 7686 10998 17257 5436 880 11835 11348
23184 10869 28263 32093 7947 29058 10759 11633 9781 21778 22132 23569 21211 10438 17565 29444 4605 26296 774 4169 27879
2317 8465 19536 1693 8649 14205 12394 2172 9447 23084 29053 16927 6314 22313 17148 128 8238 26345 22038 24671 5186 12753
22830 8168 7802 20217 2956 13200 2687 8240 14674 8768 16810 12096 13233 27829 9425 28787 6492 1063 14433 3669 829 5160
22189 8892 26984 9168 23686 262 21581 21015 1036 30114 11825 12674 25027 7890 18655 1794 14314 17070 27546 28697 5128 18029
32087 13374 27335 10976 7977 17795 9733 12863 19197 13429 22825 4904 19079 6979 4019 23781 6780 17443 27160 28910 3586 3826
31948 13742 18021 23066 9632 26250 21259 29369 19278 9504 3498 20480 15186 12236 1720 7641 6652 24945 5667 23244 6340 24182
25667 3363 26333 4878 32525 21324 4109 26301 20361 21376 15803 18712 902 20130 3658 31692 16268 23146 30373 15841 11538
1730 25115 15424 14577 29296 1134 27507 5657 4854 4646 11949 22658 13238 21291 28682 15462 28762 17299 3607 28078 11891
14009 17045 25514 23136 12431 26234 9476 27506 26933 14137 2039 29518 11978 323 17084 15506 6530 21454 24645 15547 17912
8661 27950 23878 2502 29792 28913 24597 22100 4613 1881 25365 15853 16303 11833 18206 20248 32414 7005 32282 14885 27567
11969 1130 29958 9182 21761 2437 20832 21001 28177 3042 14562 18935 3118 27673 982 21656 7956 15173 7669 4494 24434 18604
28032 4589 3739 12962 12428 4159 24190 32426 17566 27084 5592 31967 3191 15640 17576 31378 26673)

Employee 2: 5505329 pages (20908 7396 2101 29579 11545 17258 22135 32386 31035 11577 28337
29729 8123 23777 19073 9113 16087 5176 3764 8084 23398 20047 28010 10183 29718 5540 29556 20678 23033 2713 17665 24093

29248 11813 3450 668 17734 17610 9898 31474 32464 26055 15138 13950 17299 2702 26588 609 32758 12920 8478 27352 5146 28344
24225 18121 279 27278 21235 8603 31625 17341 12826 30539 1868 18238 19322 30396 131 26166 31481 15951 13791 30502 13664
18100 30750 5985 13316 28672 17105 16580 935 25855 25187 26529 31385 18635 15191 13564 23074 25245 5841 12538 7973 15332
6232 29078 17917 25748 16602 8598 1451 20929 8944 29629 17971 2668 11364 22083 22083 5789 10481 31362 9238 25778 25161 1266
32152 6390 349 31063 7115 31605 5336 14137 32105 30735 24812 6805 21707 25804 23634 5330 3337 623 32396 20982 10851 27470
17262 15898 2527 25696 2012 32540 17460 17560 9173 13971 14311 26225 26033 8200 32436 22475 21158 24419 21039 15349 20625
5027 9131 5131 7411 466 27483 24962 13626 27704 29367 22273 5187 27088 7713 2930 26908 20984 31177 25434 11022 4204 3534
9317 11995 30069 15782 26438 8516 22110 14015 23485 2766 4648 12902 4857 16582 20095 3817 17441 26499 236 1016 16174 27115
6219 30289 507 2617 23168 11933 9119 28669 19046 30612 2554 22371 10760 26472 7566 17849 27924 22625 28560 26412 12172
21700 1599 31186 1215 31226 17269 27118 12920 5870 6524 879 11381 2487 546 23737 25626 8311 27664 24232 16487 387 4695 3538
21032 24260 29260 30989 6251 30813 21398 28978 26935 12941 5765 28192 21115 17111 15526 17667 1861 14433 24468 5078 11572
25499 2576 1574 7728 24864 10068 22297 29272 19425 31909 13488 26172 23168 24872 26697 29005 27642 10230 26728 8338 18313
28786 14262 2460 5065 20699 13642 27067 18713 3869 16936 17778 9028 23533 24583 7209 22113 17428 11608 1058 2807 19467
25219 17070 25957 27081 13428 13473 7297 20697 16470 6915 25472 17583)

Employee 3: 5502185 pages (25758 8200 11030 25916 11084 10306 32221 10688 10605 17430
31739 32224 30693 21869 14848 13560 10603 15787 1805 18354 25045 10213 24123 4056 29334 17231 1850 30310 8325 499 10809
18404 30432 12201 23302 8946 19344 30019 24439 13605 22702 7176 13576 14865 3911 9572 6597 28839 1884 2781 7275 31426 12589
14921 30666 23292 11192 8766 15725 1222 25724 25291 4989 12984 25730 30309 2531 30501 28318 26404 19117 15091 21235 25648
10675 7849 7093 19318 22394 29130 30515 31647 15192 13410 788 578 14553 22238 23776 31320 25698 11296 7107 12017 27412
16851 6030 22054 12787 4860 8447 30799 28744 18642 3906 22390 18756 32208 18103 27403 10469 14088 7414 17368 26654 1978
15889 27704 31794 4901 30489 6276 13665 24733 2531 15311 20678 13790 28171 21715 24960 27552 9018 1834 21776 28780 30885
21698 5890 28077 28118 2955 6096 17162 4271 15218 15107 19443 31916 18579 28827 9082 24424 28074 32177 23141 9827 17099
14427 21377 14850 7252 15056 14905 27095 22047 20940 18278 28195 24316 10806 18537 8857 6238 513 7288 24165 24148 24656
7221 3437 19387 6365 8069 28375 3763 7835 5535 7371 24911 27034 25443 16762 31377 29739 31726 7519 2965 4934 26475 17092
7436 11816 11307 10889 24812 6464 19961 1569 19558 6220 22439 29405 3609 19392 8941 29293 13445 3431 21108 19620 6338 76
12845 15995 24945 22491 29159 6745 12788 6724 28193 15761 23002 25120 31516 32556 6301 25431 26165 31722 18832 14145 8562
25475 3091 13323 19803 11058 15369 21341 5387 18804 20108 27705 4230 24334 26335 17447 8525 29005 14243 15247 32048 6545
1102 6502 23396 19981 22234 14860 30677 10184 15802 9589 24765 18919 9746 20616 17014 20015 9767 3524 9613 23238 19477
28969 21271 17721 17600 11094 21320 26307 28428 8175 14913 1485 12113 15169 14122 30897 5897 19241 6529 26051 13516 24369
16561 17554 8319 29693 10727 1131 6088 28134 4129 31775 16068 9956 21128 1798 22781 5041 5375 29508 3734)

The optimal division of books to minimize the maximum
workload is: 5517714 pages

Time: 0.015 s

Test Number 3:

Input:

3

9

500 300 200 600 700 400 100 800 900

Output:

The optimal division of books to minimize the maximum workload is:

Employee 1: 1600 pages (500 300 200 600)

Employee 2: 1200 pages (700 400 100)

Employee 3: 1700 pages (800 900)

The optimal division of books to minimize the maximum workload is: 1700 pages

Time: 0.016 s

Test Number 4:

Input:

3

12

250 100 150 300 200 350 400 450 500 550 600 650

Output:

The optimal division of books to minimize the maximum workload is:

Employee 1: 1750 pages (250 100 150 300 200 350 400)

Employee 2: 1500 pages (450 500 550)

Employee 3: 1250 pages (600 650)

The optimal division of books to minimize the maximum workload is: 1750 pages

Time: 0.016 s

Test Number 5:

Input:

3

9

1000 2000 3000 4000 5000 6000 7000 8000 9000

Output:

The optimal division of books to minimize the maximum workload is:

Employee 1: 15000 pages (1000 2000 3000 4000 5000)

Employee 2: 13000 pages (6000 7000)

Employee 3: 17000 pages (8000 9000)

The optimal division of books to minimize the maximum workload is: 17000 pages

Time: 0.015 s

Test Number 6:

Input:

3

12

123 234 345 456 567 678 789 890 901 1012 1123 1234

Output:

The optimal division of books to minimize the maximum workload is:

Employee 1: 3192 pages (123 234 345 456 567 678 789)

Employee 2: 2803 pages (890 901 1012)

Employee 3: 2357 pages (1123 1234)

The optimal division of books to minimize the maximum workload is: 3192 pages

Time: 0.014 s

Test Number 7:

Input:

3

12

100 50 25 75 150 125 175 200 225 250 275 300

Output:

The optimal division of books to minimize the maximum workload is:

Employee 1: 700 pages (100 50 25 75 150 125 175)

Employee 2: 675 pages (200 225 250)

Employee 3: 575 pages (275 300)

The optimal division of books to minimize the maximum workload is: 700 pages

Time: 0.015 s

Test Number 8:

Input:

3

12

100 200 300 400 500 600 700 800 900 1000 1100 1200

Output:

The optimal division of books to minimize the maximum workload is:

Employee 1: 2800 pages (100 200 300 400 500 600 700)

Employee 2: 2700 pages (800 900 1000)

Employee 3: 2300 pages (1100 1200)

The optimal division of books to minimize the maximum workload is: 2800 pages

Time: 0.015 s

Test Number 9:

Input:

3

15

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Output:

The optimal division of books to minimize the maximum workload is:

Employee 1: 36 pages (1 2 3 4 5 6 7 8)

Employee 2: 42 pages (9 10 11 12)

Employee 3: 42 pages (13 14 15)

The optimal division of books to minimize the maximum workload is: 42 pages

Time: 0.015 s

Test Number 10:

Input:

3

10

500 1500 2500 3500 4500 5500 6500 7500 8500 9500

Output:

The optimal division of books to minimize the maximum workload is:

Employee 1: 18000 pages (500 1500 2500 3500 4500 5500)

Employee 2: 14000 pages (6500 7500)

Employee 3: 18000 pages (8500 9500)

The optimal division of books to minimize the maximum workload is: 18000 pages

Time: 0.014 s

Number Of Books	Execution Time (ms)
9	14
10	14
12	15
15	15
1000	15
2000	16
3000	15
4000	19
16000	20
112000	53

It is observed that the Execution Time (ms) remains constant regardless of the Number Of Books we have.

$$T(n)=O(n \log(\text{sum}))$$