**LAPORAN PRAKTIKUM 4**

**ANALISIS ALGORITMA**



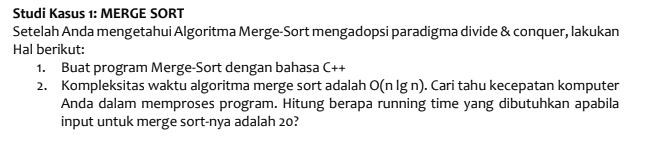
# Dibuat oleh:

AHMAD IRFAN FADHOLI 140810180034

**UNIVERSITAS PADJADJARAN**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

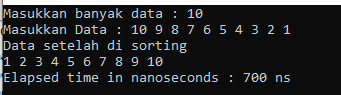
**2020**

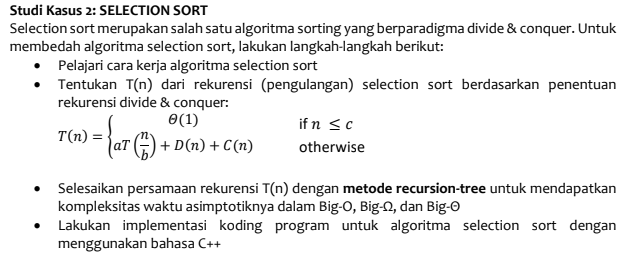


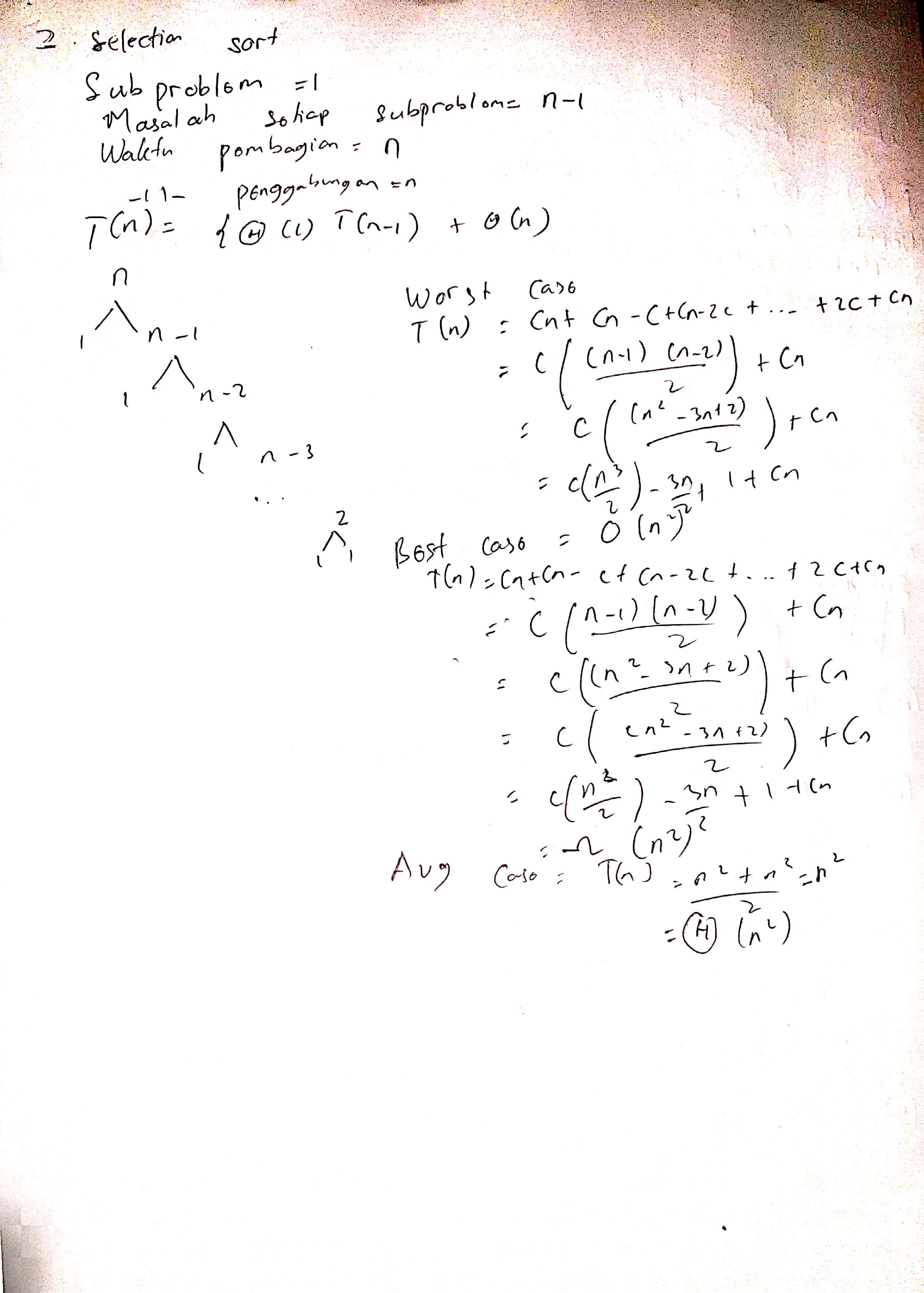
1. #include <iostream>
2. #include <chrono>
3. #include <unistd.h>
4. **using** **namespace** std;
5. **void** merge(**int** \*,**int**, **int** , **int** );
6. **void** merge\_sort(**int** \*arr, **int** low, **int** high){
8. **int** mid;
9. **if** (low < high){
11. mid=(low+high)/2;
12. merge\_sort(arr,low,mid);
13. merge\_sort(arr,mid+1,high);
14. merge(arr,low,high,mid);
15. }
16. }
18. **void** merge(**int** \*arr, **int** low, **int** high, **int** mid){
19. **int** i, j, k, c[50];
20. i = low;
21. k = low;
22. j = mid + 1;
23. **while** (i <= mid && j <= high) {
24. **if** (arr[i] < arr[j]) {
25. c[k] = arr[i];
26. k++;
27. i++;
28. }
29. **else**  {
30. c[k] = arr[j];
31. k++;
32. j++;
33. }
34. }
35. **while** (i <= mid) {
36. c[k] = arr[i];
37. k++;
38. i++;
39. }
40. **while** (j <= high) {
41. c[k] = arr[j];
42. k++;
43. j++;
44. }
45. **for** (i = low; i < k; i++)  {
46. arr[i] = c[i];
47. }
48. }

51. **int** main(){
52. **int** arr[20], num;
53. cout<<"Masukkan banyak data : ";
54. cin>>num;
55. cout<<"Masukkan Data : ";
56. **for** (**int** i = 0; i < num; i++) {
57. cin>>arr[i];
58. }
59. auto start = std::chrono::steady\_clock::now();
60. merge\_sort(arr, 0, num-1);
61. auto end = std::chrono::steady\_clock::now();
62. auto elapsed =
63. std::chrono::duration\_cast<std::chrono::nanoseconds>(end - start);
65. cout<<"Data setelah di sorting\n";
66. **for** (**int** i = 0; i < num; i++){
67. cout<<arr[i]<<" ";
68. }
69. cout << "\nElapsed time in nanoseconds : " << elapsed.count()<< " ns" << endl;
71. }

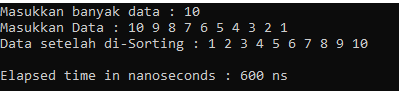
Kompleksitas waktu :

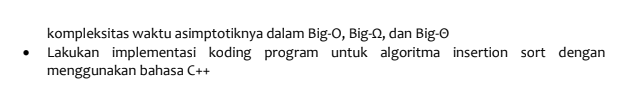
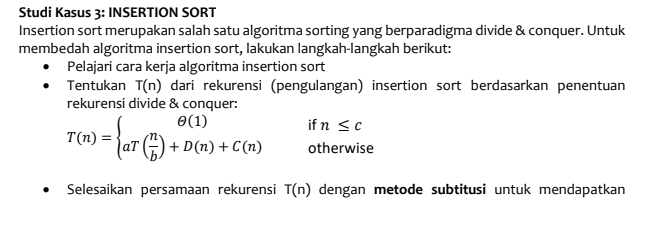


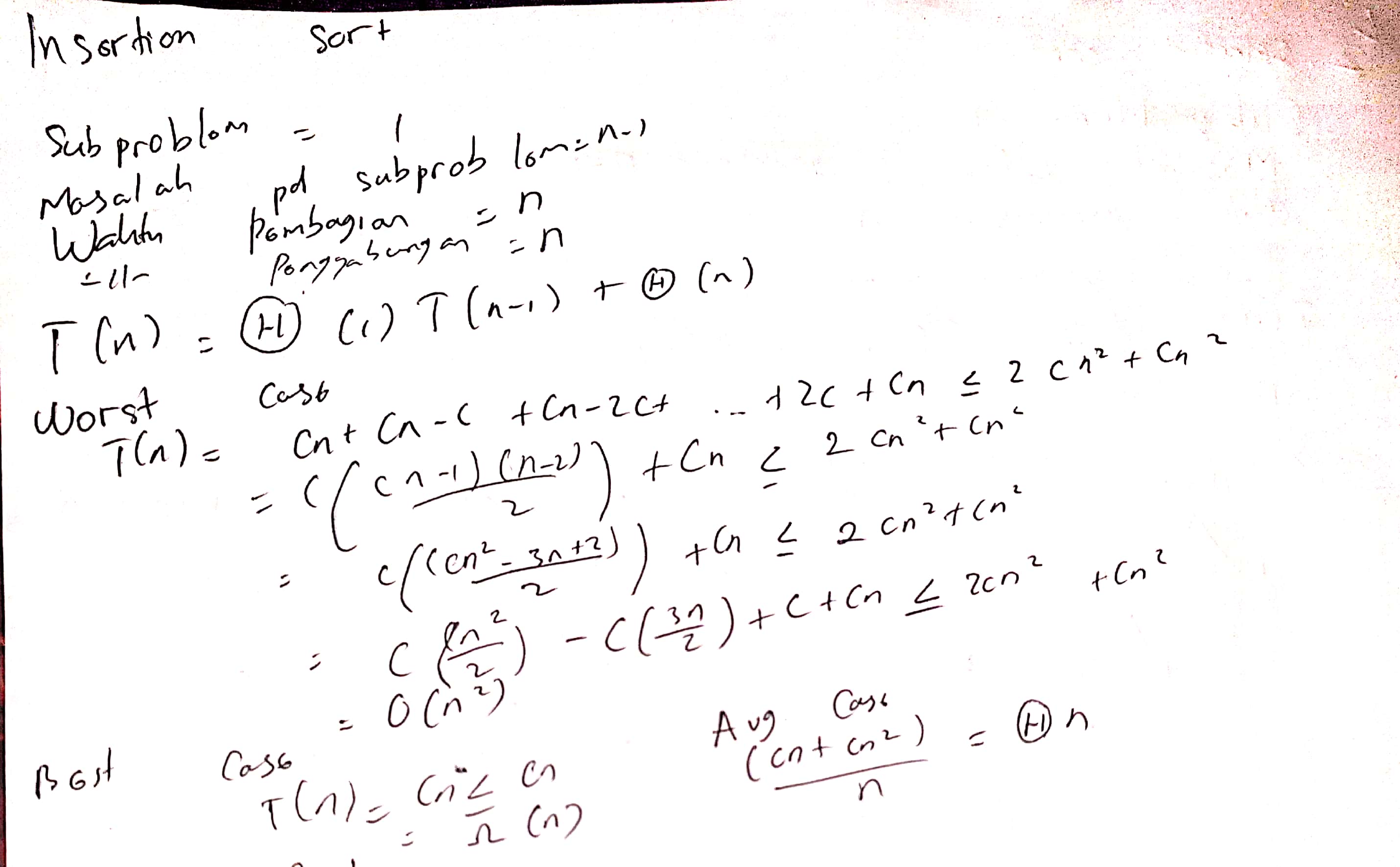




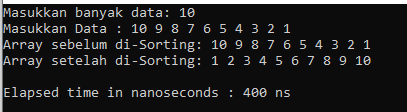
1. #include<iostream>
2. #include <chrono>
3. #include <unistd.h>
4. **using** **namespace** std;
5. **void** swap(**int** &a, **int** &b) {
6. **int** temp;
7. temp = a;
8. a = b;
9. b = temp;
10. }
11. **void** printData(**int** \*array, **int** length) {
12. **for**(**int** i = 0; i<length; i++)
13. cout << array[i] << " ";
14. cout << endl;
15. }
16. **void** selectionSort(**int** \*array, **int** length) {
17. **int** i, j, imin;
18. **for**(i = 0; i<length-1; i++) {
19. imin = i;
20. **for**(j = i+1; j<length; j++)
21. **if**(array[j] < array[imin])
22. imin = j;
23. swap(array[i], array[imin]);
24. }
25. }
26. **int** main() {
27. **int** n,arr[30];
28. cout << "Masukkan banyak data : ";
29. cin >> n;
30. cout << "Masukkan Data : ";
31. **for**(**int** i = 0; i<n; i++) {
32. cin >> arr[i];
33. }
34. auto start = std::chrono::steady\_clock::now();
35. selectionSort(arr, n);
36. auto end = std::chrono::steady\_clock::now();
37. auto elapsed = std::chrono::duration\_cast<std::chrono::nanoseconds>(end - start);
38. cout << "Data setelah di-Sorting : ";
39. printData(arr, n);
40. cout << "\nElapsed time in nanoseconds : " << elapsed.count()<< " ns" << endl;
41. }

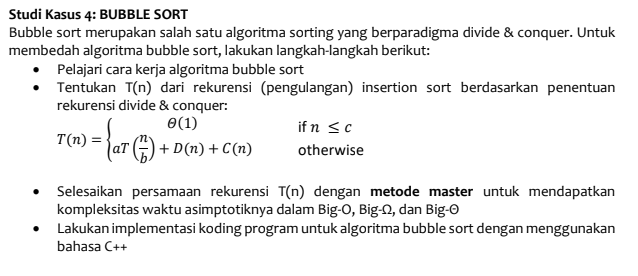


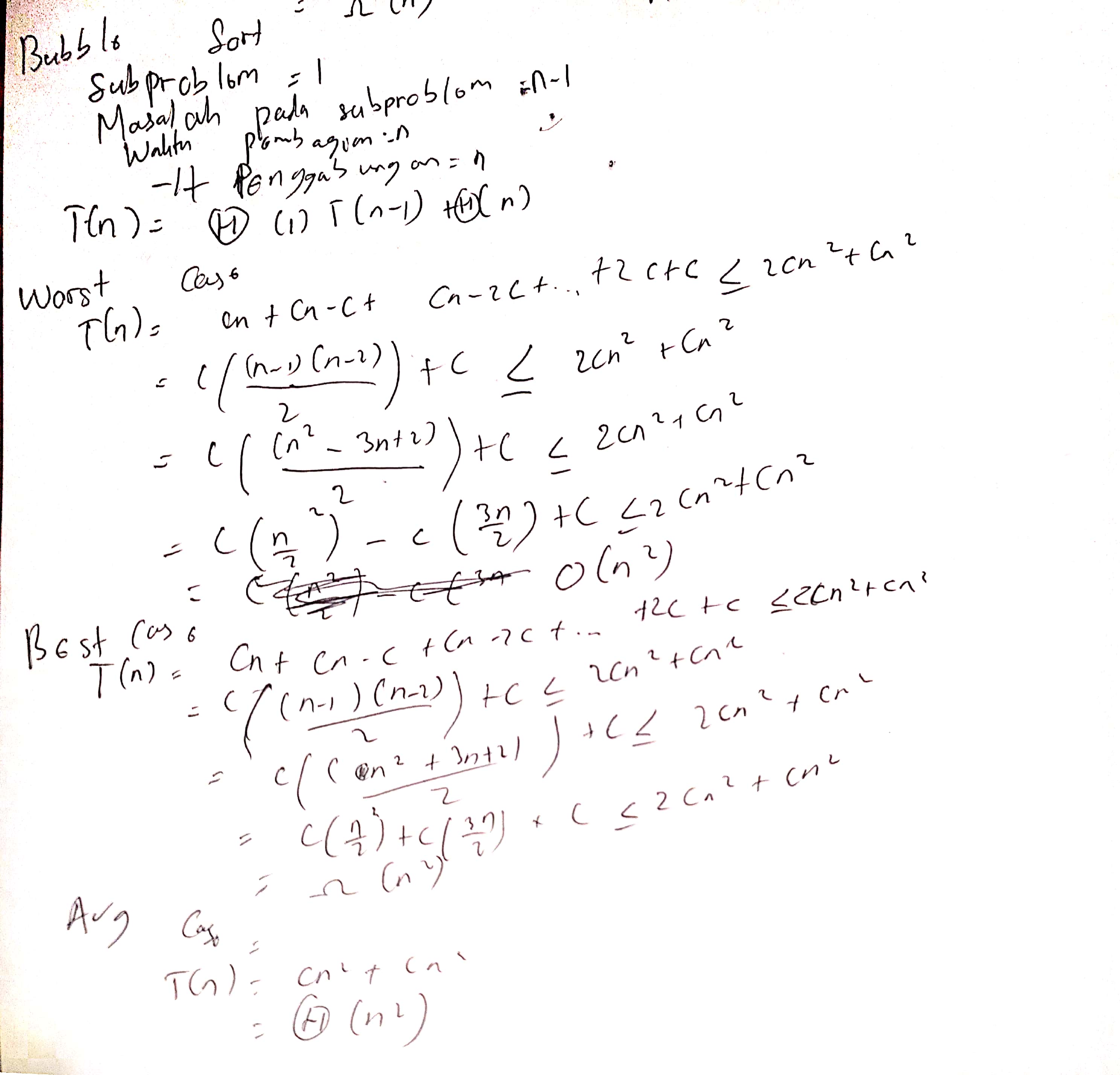




1. #include<iostream>
2. #include <chrono>
3. #include <unistd.h>
4. **using** **namespace** std;
5. **void** insertionSort(**int** \*array, **int** length) {
6. **int** temp, j;
7. **for**(**int** i = 1; i<length; i++) {
8. temp = array[i];
9. j = i;
10. **while**(j > 0 && array[j-1]>temp) {
11. array[j] = array[j-1];
12. j--;
13. }
14. array[j] = temp;
15. }
16. }
17. **void** printData(**int** \*array, **int** length) {
18. **for**(**int** i = 0; i<length; i++)
19. cout << array[i] << " ";
20. cout << endl;
21. }
23. **int** main() {
24. **int** n,arr[30];
25. cout << "Masukkan banyak data: ";
26. cin >> n;
28. cout << "Masukkan Data : ";
29. **for**(**int** i = 0; i<n; i++) {
30. cin >> arr[i];
31. }
32. cout << "Array sebelum di-Sorting: ";
33. printData(arr, n);
34. auto start = std::chrono::steady\_clock::now();
35. insertionSort(arr, n);
36. auto end = std::chrono::steady\_clock::now();
37. auto elapsed = std::chrono::duration\_cast<std::chrono::nanoseconds>(end - start);
38. cout << "Array setelah di-Sorting: ";
39. printData(arr, n);
40. cout << "\nElapsed time in nanoseconds : " << elapsed.count()<< " ns" << endl;
41. }







1. #include<iostream>
2. #include <chrono>
3. #include <unistd.h>
4. **using** **namespace** std;
6. **void** swap(**int** &a, **int** &b) {
7. **int** temp;
8. temp = a;
9. a = b;
10. b = temp;
11. }
12. **void** printData(**int** \*array, **int** length) {
13. **for**(**int** i = 0; i<length; i++)
14. cout << array[i] << " ";
15. cout << endl;
16. }
17. **void** bubbleSort(**int** \*array, **int** length) {
18. **for**(**int** i = 0; i<length; i++) {
19. **int** isSwap = 0;
20. **for**(**int** j = 0; j<length-i-1; j++) {
21. **if**(array[j] > array[j+1]) {
22. swap(array[j], array[j+1]);
23. isSwap = 1;
24. }
25. }
26. **if**(!isSwap)
27. **break**;
28. }
29. }
30. **int** main() {
31. **int** n,arr[30];
32. cout<< "Bubble Sort\n";
33. cout << "Masukkan banyak Data: ";
34. cin >> n;
35. cout << "Masukkan Data : ";
36. **for**(**int** i = 0; i<n; i++) {
37. cin >> arr[i];
38. }
39. auto start = std::chrono::steady\_clock::now();
40. bubbleSort(arr, n);
41. auto end = std::chrono::steady\_clock::now();
42. auto elapsed = std::chrono::duration\_cast<std::chrono::nanoseconds>(end - start);
43. cout << "Array setelah di-Sorting: ";
44. printData(arr, n);
45. cout << "\nElapsed time in nanoseconds : " << elapsed.count()<< " ns" << endl;
46. }

