The Euclidean traveling salesperson problem is:
input: a positive integer n and a list P of n distinct points representing vertices of a Euclidean graph
output: a list of n points from P representing a Hamiltonian cycle of minimum total weight for the graph.

Exhaustive Optimization

```
// calculate the farthest pair of vertices
Dist = farthest(n, P);      - n^2
bestDist = n*Dist;          - 1+1

// populate the starting array for the permutation algorithm
A = new int[n];             1
// populate the array A with the values in the range 0 .. n-1
for (i = 0; i<n; i++)       } n
    A[i] = i;

// calculate the Hamiltonian cycle of minimum weight
print_perm(n, A, n, P, bestSet, bestDist);      n * n!
```

$n^2+3+n+n*n!$

$n*n! + n^2+n+3 \in O(n*n!)$

$n*n! + n^2+n+3 \leq c\, n\, n! \;\; \forall n > n_0$

let $C=6$

$n*n! + n^2+n+3 \leq 6n\, n! \;\; \forall n > n_0$

$n^2+n+3 \leq 5n \cdot n! \;\; \forall n > n_0$

let $n_0 = 1$

$4 \leq 5$ true $\forall n > 1$

therefore

$n*n! + n^2+n+3 \in O(n*n!)$

```
//P is the set of points
//A is the array in indexes
void print_perm(int n, int *A, int sizeA, point2D *P, int *bestSet, float &bestDist)
// function to generate the permutation of indices of the list of points
{
    int i; float dist, total = 0;       1+1+1

    if (n == 1) {
        // we obtain a permutation so we compare it against the current shortest
        float dist = 0;     1
        for (int i = 0; i < n; i++) {
                 6   1   6   1   6   1   6
            dist = (P[A[i]].x - P[A[i + 1]].x)*(P[A[i]].x - P[A[i + 1]].x) + (P[A[i]].y - P[A[i + 1]].y)*(P[A[i]].y - P[A[i + 1]].y);   } 30n
            dist = sqrt(dist);   1
            total += dist;   1+1
31n+31
        }
        //add the edge of starting A and ending point A to complete a cycle
                 6   1   6   1   6   1   6
        dist = (P[A[0]].x - P[A[n-1]].x)*(P[A[0]].x - P[A[n-1]].x) + (P[A[0]].y - P[A[n-1]].y)*(P[A[0]].y - P[A[n-1]].y);   } 30
        dist = sqrt(dist);
        total += dist;

        //found a shorter hamiltonian cycle
        if (total < bestDist) {
            //update bestDist
            bestDist = total;   1
                                        max(n+1, 0) = n+1
            //add A indexes values to bestSet found a better set~
            for (int i = 0; i < n; i++) {
                bestSet[i] = A[i];   } n
            }
        }


}
```

```
        else {
                for (i = 0; i< n - 1; i++) {
                        print_perm(n - 1, A, sizeA, P, bestSet, bestDist);   n!
                        if (n % 2 == 0) {
                                // swap(A[i], A[n-1])
                             1  int temp = A[i];
                   5    1+1     A[i] = A[n - 1];
                        1+1     A[n - 1] = temp;
                        }
                        else
                        {
                                // swap(A[0], A[n-1])
                             1  int temp = A[0];
                   5    1+   1  A[0] = A[n - 1];
                        1+1     A[n - 1] = temp;
                        }
                }
                print_perm(n - 1, A, sizeA, P, bestSet, bestDist);   n!
        }
}
```

max (5, 5) = 5

(n-1-0+1) * (n! + 5)

n * n! + 5n

else branch = n*n! + n! + 5n

if branch = 31n + 31

R.t. Print_Perm = 3 + max(31n + 31, n*n! + n! + 5n)

= n*n! + n! + 5n + 3

$n*n! + n! + 5n + 3 \in O(n*n!)$

$$\lim_{n\to\infty} \frac{n*n! + n! + 5n + 3}{n*n!} \Rightarrow \lim_{n\to\infty} \frac{1 + \cancel{1/n} + 5/n! + 3/n*n!}{1}$$

$$\lim_{n\to\infty} \frac{1 + \cancel{1/n}^{\to 0} + \cancel{5/n!}^{\to 0} + \cancel{3/n\,n!}^{\to 0}}{1} = 1 > 0 \text{ and constant}$$

$\boxed{\text{therefore R.t Print\_Perm} = n*n!}$

$\boxed{\text{R.t Exhaustive} = n*n!}$

C:\WINDOWS\system32\cmd.exe

```
CPSC 335-x - Programming Assignment #3
Euclidean traveling salesperson problem: exhaustive optimization algorithm
Enter the number of vertices (>2)
3
Enter the points; make sure that they are distinct
x=5
y=6
x=1
y=2
x=7
y=8
The Hamiltonian cycle of the minimum length
Point [0] = (7 , 8) , Point [1] = (1 , 2) , Point [2] = (7 , 8) , Point [3] = (7 , 8),
Minimum length is 2.82843
elapsed time: 0 seconds
Press any key to continue . . .
```

C:\WINDOWS\system32\cmd.exe

```
CPSC 335-x - Programming Assignment #3
Euclidean traveling salesperson problem: exhaustive optimization algorithm
Enter the number of vertices (>2)
5
Enter the points; make sure that they are distinct
x=1
y=1
x=2
y=2
x=3
y=3
x=4
y=4
x=5
y=5
The Hamiltonian cycle of the minimum length
Point [0] = (1 , 1) , Point [1] = (2 , 2) , Point [2] = (3 , 3) , Point [3] = (4 , 4) , Point [4] = (5 , 5) , Point [5] = (1 , 1),
Minimum length is 1.41421
elapsed time: 0 seconds
Press any key to continue . . .
```

C:\WINDOWS\system32\cmd.exe

```
CPSC 335-x - Programming Assignment #3
Euclidean traveling salesperson problem: exhaustive optimization algorithm
Enter the number of vertices (>2)
4
Enter the points; make sure that they are distinct
x=2
y=5
x=3
y=4
x=8
y=9
x=1
y=1
The Hamiltonian cycle of the minimum length
Point [0] = (2 , 5) , Point [1] = (3 , 4) , Point [2] = (8 , 9) , Point [3] = (1 , 1) , Point [4] = (2 , 5),
Minimum length is 1.41421
elapsed time: 0 seconds
Press any key to continue . . .
```