

IF2211 Strategi Algoritma

Tugas Kecil I : Convex Hull dengan Kompleksitas Waktu $O(n^3)$



Oleh:

Matthew Kevin Amadeus 13518035

**PROGRAM STUDI SARJANA INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2020**

BAB I

ALGORITMA DAN KOMPLEKSITAS

1. Definisi Convex Hull

Secara singkat, *convex hull* adalah poligon *convex* yang dihasilkan dari beberapa titik di sebuah bidang, di mana poligon tersebut melingkupi semua titik yang ada di bidang tersebut sehingga sedemikian untuk setiap dua titik berbeda yang dibuat garis antara kedua garis itu, tidak ada garis yang memotong ke luar poligon tersebut.

2. Algoritma pencarian *Convex Hull* dari sejumlah titik

Terdapat beberapa cara untuk mendapatkan *convex hull* dari beberapa titik, dan pada tugas kali ini penulis ditugaskan untuk membuat *convex hull* dengan cara *brute-force*. Beberapa analisis penting yang diambil oleh penulis untuk menyusun algoritma *brute-force* ini adalah sebagai berikut:

1. Perlu diperhatikan bahwa titik-titik ekstrim (titik dengan absis atau ordinat maksimum dan minimum) dari koleksi titik sudah pasti masuk ke dalam *convex hull*.
2. Sifat dari sisi poligon yang dibentuk yakni ketika sisi tersebut diperpanjang menjadi sebuah persamaan garis lurus adalah semua titik lain akan berada sepenuhnya di sebelah kanan atau sepenuhnya sebelah kiri, dan mungkin tepat berada di dalam garis.

Algoritma *brute-force* yang dibuat adalah sebagai berikut:

1. Cari titik ekstrim yang berada dalam koleksi titik tersebut. Sebagai konvensi, penulis akan memilih untuk memilih titik yang paling kiri (memiliki nilai x minimum).
2. Secara *brute-force*, cari titik berbeda dan buat persamaan garisnya.
3. Dari persamaan garis yang dibuat, lakukan pengecekan terhadap semua titik, apakah semua titik lain berada di satu sisi yang sama dari garis tersebut.
4. Bila ditemukan, maka titik tersebut menjadi bagian dari *convex hull*.
5. Ulangi langkah 2-4 hingga kembali ke titik asal.

3. Kompleksitas Program

Dengan algoritma seperti tertera, kompleksitas waktunya adalah $O(n^3)$. Kompleksitas waktu tersebut berasal dari jumlah loop yang ada, karena akan mencari dengan semua titik.

BAB II

KODE PROGRAM

Berikut ini adalah implementasi dari program yang dibuat oleh penulis. Penulis membuat program ini dengan bahasa C++ dengan bantuan library FreeGLUT untuk visualisasinya. Selain itu, spesifikasi komputer yang digunakan oleh penulis adalah sebagai berikut:

- a. Processor : Intel(R) Core(TM) i7-8550U @ 1.80GHz
- b. Memory : 8192 MB
- c. GPU : Intel(R) UHD Graphics 620 dan NVIDIA GeForce MX150

Berikut ini adalah screenshot dari program yang telah dibuat.

1. Variabel Global dan *Include* yang dibutuhkan

```
#include <GL/glut.h>
#include <windows.h>
#include <bits/stdc++.h>
#include <chrono>

#define OFFSET_X 0
#define OFFSET_Y 0

using namespace std;

vector<pair<int,int>>
vpointkmpair<int,int> result;
```

2. Fungsi utama (*main*)
 - a. Input titik dan pilihan untuk mengacak titik

```
// Initialization
int n;
char modeInput;

// Initialize seed for pseudo-random number generator
srand (time(NULL));

std::printf("Randomize input?(y/n): ");
scanf("%c", &modeInput);
bool randomize = (modeInput=='y' || modeInput == 'Y'
);
std::printf("Enter points count: ");
scanf("%d", &n);

for(int i=1;i<=n;i++)
{
    int x,y;
    if(randomize)
    {
        x = rand() % 151;
        y = rand() % 151;
    }
    else
    {
        std::printf("Enter point %d: ", i);
        scanf("%d %d", &x, &y);
    }
    pointList.push_back({x,y});
}

printPointList(pointList);
```

b. Algoritma Convex Hull dan Penghitungan Eksekusinya

```
// Start Timer
auto start = std::chrono::steady_clock::now();

// Convex Hull 1: Find furthest Left point
int pivotIndex=0;
// Find minimum index
for(int i=1;i<pointList.size();i++) if(pointList[pivotIndex].first > pointList[i].first) pivotIndex = i;

pair<int,int> pivot = pointList[pivotIndex];
// Convex Hull 2 : Making the convex hull itself

// Store the starting index
int startingIndex = pivotIndex;
int lastIndex;
// Loop until meets the starting index
int count=0;
do
{
    int i=0;
    bool found=false;
    while(!found && i<n)
    {
        // If a Line is created (two different points are selected)..
        if(i!=pivotIndex)
        {
            // Constructing the Line equation
            int x1 = pointList[i].first, y1 = pointList[i].second;
            int x2 = pivot.first , y2 = pivot.second;
            int a = y2-y1, b = x1-x2, c = x1*y2-y1*x2;

            // Check if onesided
            bool oneSided = true;
            int last = 0;
            int j=0;
            while(oneSided && j<n)
            {
                if(j!=i && j!=pivotIndex)
                {
                    int x = pointList[j].first, y=pointList[j].second;
                    int calc = a*x + b*y - c;

                    if((last==1 && calc<0) || (last==-1 && calc>0)) oneSided=false;
                    else if(last==0) last = (calc>0) ? 1 : -1; // First point found
                }

                j++;
            }

            if(lastIndex!=i && oneSided)
            {
                result.push_back(pointList[i]);
                lastIndex = pivotIndex;
                pivot = pointList[i];
                pivotIndex = i;
                found = true;
            }
        }
        i++;
    }
} while(pivotIndex!=startingIndex);

auto stop = std::chrono::steady_clock::now();
auto duration = std::chrono::duration_cast<std::chrono::nanoseconds>(stop-start);
```

c. Fungsi untuk menampilkan output teks dan visualisasi

```
std::cout << '\n' << "Time taken: " << duration.count() << " ns\n";

std::printf("Convex Hull:\n");
printPointList(result);

int argc = 1;
glutInit(&argc, NULL);

glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(800,800);
glutInitWindowPosition(100, 100);

glutCreateWindow("Convex Hull");
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluOrtho2D(-20.0, 170.0, -20.0, 170.0);

glutDisplayFunc(render);
glutReshapeFunc(resize);

glutMainLoop();
```

3. Fungsi untuk *debugging*

```
// Debugging functions
void printPoint(pair<int,int> p)
{
    std::printf("<%d,%d>", p.first, p.second);
}

void printPointList(vector<pair<int,int>> v)
{
    for(vector<pair<int,int>>::iterator ii=v.begin(); ii!=v.end(); ii++)
    {
        printPoint(*ii);
        std::printf(" ");
    }
    printf("\n");
}
```

4. Fungsi untuk *render* visualisasi *convex hull*

```
void render(void)
{
    glClear(GL_COLOR_BUFFER_BIT); // clear the drawing buffer.
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH | GLUT_MULTISAMPLE);
    glEnable(GLUT_MULTISAMPLE);

    // Make Convex Hull Lines
    glLineWidth(0.5f);
    glColor3f(1.0, 0.941, 0.0);
    glEnable(GL_LINE_SMOOTH);
    int prevX=result[0].first,prevY=result[0].second;
    for(int i=0;i<result.size();i++)
    {
        // Draw Line for the convex hull result
        glBegin(GL_LINES);
        glVertex2i(prevX + OFFSET_X,prevY + OFFSET_Y);
        glVertex2i(result[(i+1)%result.size()].first + OFFSET_X,result[(i+1)%result.size()].second + OFFSET_Y);
        glEnd();

        // Change previous point to the next pivot
        prevX=result[i+1].first;
        prevY=result[i+1].second;
    }

    // Make the set of points
    glEnable(GL_POINT_SMOOTH);
    glPointSize(5.0f);
    glColor3f(0.2111, 0.5025, 0.2864);
    for(int i = 0; i < pointList.size(); i++)
    {
        // Draw point
        glBegin(GL_POINTS);
        glVertex2i(pointList[i].first + OFFSET_X,pointList[i].second + OFFSET_Y);
        glEnd();
    }

    glPointSize(7.0f);
    glColor3f(0.6676, 0.1769, 0.1555);
    for(int i = 0; i < result.size(); i++)
    {
        // Draw point
        glBegin(GL_POINTS);
        glVertex2i(result[i].first + OFFSET_X,result[i].second + OFFSET_Y);
        glEnd();
    }

    glFlush();
}
```

5. Fungsi untuk mengatur aspek dari layar visualisasi

```
void resize(int width, int height)
{
    GLfloat aspect = (GLfloat)width / (GLfloat)height;
    glViewport(0, 0, width, height);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if (width >= height) gluOrtho2D(-1.0 * aspect, 1.0 * aspect, -1.0, 1.0);
    else gluOrtho2D(-1.0, 1.0, -1.0 / aspect, 1.0 / aspect);
}
```


BAB III

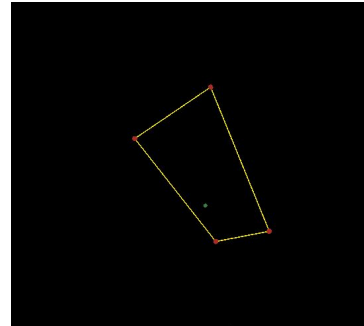
SCREENSHOT PROGRAM

Berikut ini adalah tampilan program ketika diberikan masukan yang berbeda-beda.

Untuk N=5,

```
D:\_University\Sem. 4\Stima\Convex Hull>run
Compiling...
Randomize input?(y/n): y
Enter points count: 5
<86,120> <120,36> <83,51> <42,90> <89,30>

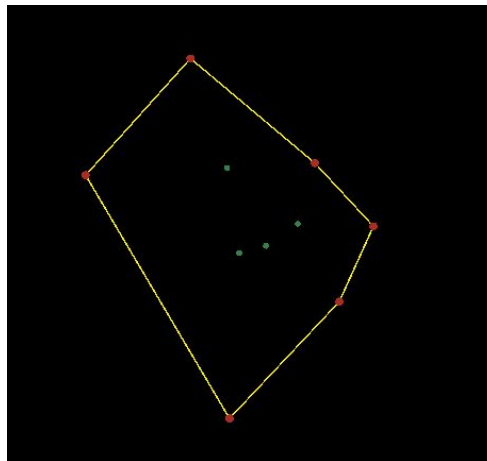
Time taken: 0 ns
Convex Hull:
<86,120> <120,36> <89,30> <42,90>
```



Untuk N=10,

```
D:\_University\Sem. 4\Stima\Convex Hull>run
Compiling...
Randomize input?(y/n): y
Enter points count: 10
<125,79> <94,80> <70,68> <50,148> <7,100> <65,103> <101,105> <66,0> <111,48> <81,71>

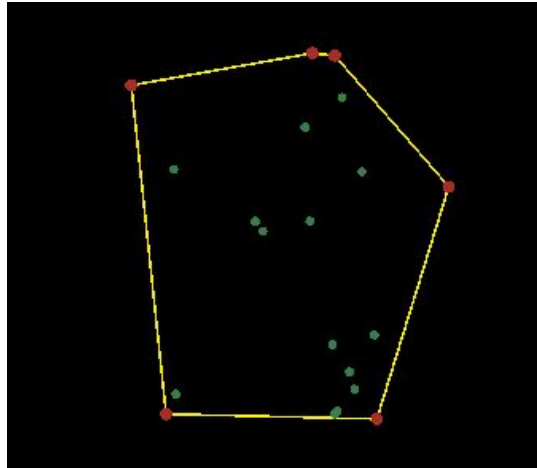
Time taken: 0 ns
Convex Hull:
<50,148> <101,105> <125,79> <111,48> <66,0> <7,100>
```



Untuk N=20,

```
D:\_University\Sem. 4\Stima\Convex Hull>run
Compiling...
Randomize input?(y/n): y
Enter points count: 20
<100,3> <31,2> <90,148> <70,76> <35,10> <17,135> <99,2> <98,30> <145,94> <110,100> <34,101> <102,130> <67,80>
<99,147> <115,34> <105,19> <87,118> <107,12> <89,80> <116,0>

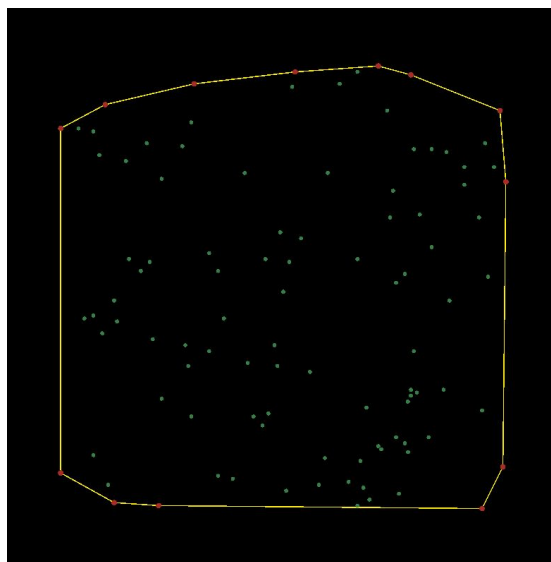
Time taken: 0 ns
Convex Hull:
<31,2> <116,0> <145,94> <99,147> <90,148> <17,135>
```



Untuk pengujian terhadap waktu, pada komputer yang penulis gunakan memang hasilnya hingga kurang lebih N=300 akan memakan waktu 0 ns.

```
D:\_University\Sem. 4\Stima\Convex Hull>run
Compiling...
Randomize input?(y/n): y
Enter points count: 100
<68,29> <120,40> <16,9> <90,114> <87,9> <107,150> <55,65> <104,4> <119,54> <148,135> <0,13> <41,123> <6,129> <44,131> <76,7> <136,110> <116,80> <130,121> <75,74> <113,25> <22,118> <112,108> <53,12> <100,2> <143,124> <114,6> <50,54> <144,79> <11,19> <142,1> <63,50> <103,35> <97,10> <124,25> <117,20> <102,8> <121,100> <18,3> <119,122> <18,71> <8,65> <29,124> <118,41> <110,135> <42,56> <11,66> <43,49> <101,17> <58,11> <14,60> <11,128> <0,129> <79,148> <33,2> <89,18> <113,77> <150,111> <141,99> <44,32> <100,148> <111,99> <116,23> <53,81> <81,92> <72,56> <73,49> <23,85> <34,38> <69,85> <125,89> <50,87> <27,81> <125,122> <146,116> <74,94> <108,21> <100,85> <78,143> <77,84> <129,41> <84,47> <30,84> <70,33> <117,37> <136,116> <19,64> <34,112> <142,34> <118,39> <107,22> <94,144> <118,147> <65,32> <31,58> <45,144> <15,137> <62,114> <131,71> <13,120> <149,15>

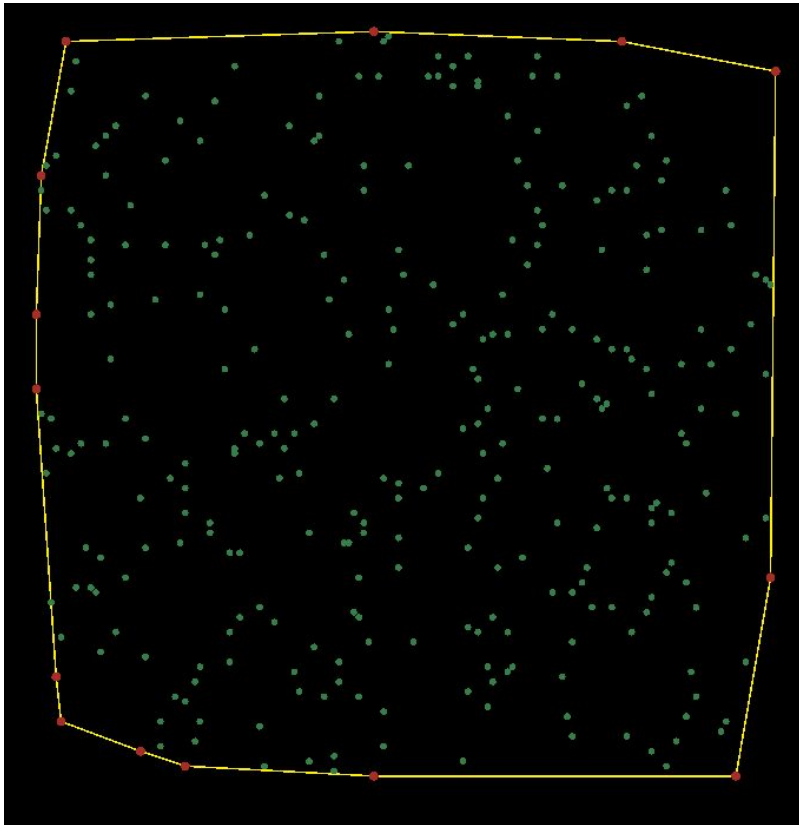
Time taken: 0 ns
Convex Hull:
<18,3> <33,2> <142,1> <149,15> <150,111> <148,135> <118,147> <107,150> <79,148> <45,144> <15,137> <0,129> <0,13>
```



Bila nilai N mencapai 300, maka waktu *runtime*-nya akan ditampilkan, dan akan berkisar sekitar 0,1ms.

```
D:\University\Sem. 4\Stima\Convex Hull>run
Compiling...
Randomize input?(y/n): y
Enter points count: 300
<89,82> <61,1> <133,21> <88,46> <33,19> <42,32> <124,82> <132,67> <51,66> <96,89> <74,106> <33,7> <55,112> <139,9> <62,23> <91,65> <67,49> <23,137> <92,22> <67,118> <109,90> <97,108> <13
5,90> <126,55> <117,118> <122,123> <64,89> <144,48> <23,46> <23,68> <39,82> <88,17> <10,111> <51,76> <22,5> <114,88> <28,60> <58,137> <53,21> <75,101> <85,91> <1,93> <98,78> <115,106> <1
03,72> <12,38> <71,13> <65,53> <95,67> <31,53> <13,37> <68,27> <64,47> <66,40> <5,125> <17,131> <61,4> <16,95> <81,99> <69,150> <119,148> <114,76> <25,96> <115,74> <56,3> <40,23> <87,93>
<35,107> <47,2> <3,61> <52,113> <107,20> <6,11> <39,94> <148,81> <95,97> <53,69> <82,61> <2,73> <1,78> <82,145> <47,117> <41,65> <88,145> <108,12> <92,14> <31,58> <14,44> <37,105> <137,
83> <62,19> <100,119> <41,66> <11,46> <30,47> <54,17> <63,100> <65,33> <56,49> <117,56> <87,3> <149,40> <12,93> <128,124> <124,109> <116,75> <120,118> <109,27> <74,56> <38,108> <66,32> <
121,84> <50,60> <146,101> <91,56> <130,7> <124,33> <31,2> <141,111> <52,131> <90,139> <127,110> <94,42> <111,39> <145,91> <42,45> <148,100> <13,127> <94,60> <87,70> <33,7> <103,111> <69,
0> <91,88> <46,67> <113,34> <66,141> <60,96> <140,118> <66,16> <104,62> <96,133> <93,89> <117,86> <102,114> <26,11> <4,35> <125,129> <72,83> <120,56> <34,128> <148,52> <36,49> <2,121> <3
7,136> <49,69> <106,141> <10,67> <99,44> <26,6> <135,74> <144,23> <49,31> <106,72> <142,0> <101,141> <43,69> <114,50> <141,86> <80,141> <103,90> <102,107> <9,38> <93,19> <142,73> <57,26>
<111,79> <61,76> <149,99> <74,48> <136,57> <62,148> <88,30> <85,143> <71,6> <8,138> <128,41> <85,139> <129,53> <93,32> <129,43> <45,86> <100,103> <46,10> <59,105> <20,115> <73,90> <74,5
9> <116,58> <105,93> <128,2> <2,118> <135,118> <127,120> <31,15> <97,22> <46,34> <90,71> <27,107> <102,145> <67,123> <57,128> <125,77> <96,29> <3,114> <31,63> <12,104> <12,108> <77,27> <
19,107> <40,45> <90,80> <109,37> <107,119> <125,36> <131,69> <90,140> <112,42> <15,129> <131,83> <5,66> <140,111> <34,22> <34,11> <134,34> <124,102> <34,97> <96,21> <12,101> <132,12> <27,
124> <71,148> <125,54> <79,58> <74,42> <76,123> <7,148> <72,149> <54,61> <106,51> <120,135> <22,56> <44,109> <30,132> <132,39> <16,84> <123,137> <59,16> <98,124> <121,29> <36,51> <19,72>
<125,51> <40,29> <90,52> <6,28> <72,94> <5,20> <109,8> <3,123> <23,24> <29,16> <58,129> <120,86> <15,67> <67,51> <92,74> <90,29> <102,130> <71,60> <15,121> <8,65> <117,34> <41,143> <120
,8> <17,29> <70,141> <105,37> <82,141> <19,40> <114,116> <14,25> <134,16> <8,114> <63,47> <9,144> <4,72> <125,5> <150,142> <57,71>

Time taken: 998000 ns
Convex Hull:
<1,78> <5,20> <6,11> <22,5> <31,2> <69,0> <142,0> <149,40> <150,142> <119,148> <69,150> <7,148> <2,121> <1,93>
```



LAMPIRAN

Poin	Ya	Tidak
Program berhasil dikompilasi		
Program berhasil <i>running</i>		
Program dapat menerima input dan menuliskan output		
Luaran sudah benar untuk semua n		