Homework 8
Shengchao Liu

1. (a) $c + g_1p_1 + \dots g_np_n + \frac{1}{2}G_{11}p_1^2 + G_{12}p_1p_2 + \dots \frac{1}{2}G_{nn} + p_n^2$

   So we have $1 + n + \frac{n(n+1)}{2} = \frac{1}{2}(n+1)(n+2)$ coefficients.

   (b) $n + 1$ points, $\frac{n(n+1)}{2}$ edges, add together we have $\frac{1}{2}(n+1)(n+2)$

   (c) G identically 0: $n + 1$

   G diagonal: $2n + 1$

   G tridiagonal: $3n$

2. $\underset{v}{min}\,\underset{p\in D_k}{max}\frac{v^Tp}{\|v\|\|p\|} = \underset{v}{min}\frac{\|v\|_\infty}{\|v\|_2} = \frac{1}{\sqrt{n}}$

3. Code is below:

```
x = [0 0; 1 0; 2 0; 1 1; 0 2; 0 1];
y = [1; 2.0084; 7.0091; 1.0168; -0.9909; -0.9916];

%% initial
n = size(x, 2);
N = (n+1)*(n+2) / 2;
big = zeros(N, N);

%% transform
for row = 1 : N
    big(row, 1) = 1;
    big(row, 2:1+n) = x(row,:);
    col = 1+n;
    for i = 1:n
        for j = i:n
            col = col + 1;
            if i == j
                big(row, col) = 0.5*x(row,i)^2;
            else
                big(row, col) = x(row,i) * x(row, j);
            end
        end
    end
end

%% calculate
solution = pinv(big) * y;

%% transform back
c = solution(1);
g = zeros(n,1);
```

```
g(1:n) = solution(2:1+n);
G = zeros(n,n);
col = n+1;
for i = 1:n
    for j = i:n
        col = col + 1;
        G(i, j) = solution(col);
        G(j, i) = solution(col);
    end
end
```

answer is below:

$c = 1$

$$g = \begin{bmatrix} -0.9878 \\ -2.9878 \end{bmatrix}$$

$$G = \begin{bmatrix} 3.9923 & 1.0000 \\ 1.0000 & 1.9923 \end{bmatrix}$$

4. code is below:

```
function [inform, x] = direct(fun, x, directparams)
global numf numg;
numf = 0;
numg = 0;
n = size(x.p, 1);
directions = [eye(n), -1*eye(n)];
gamma = 1;
x.f = feval(fun, x.p, 1);

for step = 1:directparams.maxit
    order = randperm(2*n);
    flag = 0;
    for k = 1 : 2*n
        d = directions(:, order(k));
        x_neo = struct('p', x.p + gamma*d);
        x_neo.f = feval(fun, x_neo.p, 1);
        if x_neo.f < x.f - gamma^2
            x = x_neo;
            gamma = gamma * directparams.phi;
            flag = 1;
            break;
        end
    end
    if flag == 0
        gamma = gamma * directparams.theta;
    end
    if gamma <= directparams.toler
        inform.status = 1;
        inform.iter = step;
        return;
    end
end

inform.status = 0;
inform.iter = directparams.maxit;
return
```

answer is below:

```
 Function xpowsing running BFGS
Success: 58 steps taken
  Ending point: -0.0001807 1.807e-05 0.0001787 0.0001787
     0.003425 -0.0003425 0.001808 0.001808
  Ending value: 3.139e-10 ; No. function evaluations: 112; No.
     gradient evaluations 104
  Norm of ending gradient: 4.755e-07

 Function xpowsing running Direct Search
Success: 23822 steps taken
  Ending point: -0.01508 0.001509 -0.00711 -0.007112  -0.0145
     0.001451 -0.006846 -0.006848
  Ending value: 1.885e-07 ; No. function evaluations: 276346
```