

算法设计与分析 E5-D

21377206 阮阳栋

题目描述

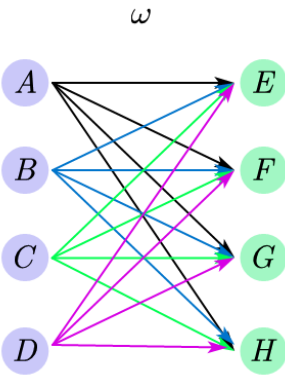
先有两组点，每组点个数为 n ，定义 $(x_i, y_i), (x'_j, y'_j)$ 两点之间的距离为 $|x_i - x'_j| + |y_i - y'_j|$ 。

现在需要将两组点进行匹配使得总距离最小，即求解：

$$\min S = \sum_{i=1}^n |x_i - x'_{p_i}| + |y_i - y'_{p_i}|$$

题目分析

可以发现这是带权二分匹配问题，并不能直接使用传统的最大流和匈牙利算法去解决，所以可以考虑更加优化的方法。



题目求解

带权二分匹配采用KM算法：

```

void match(ll u){
    ll x, y=0, newy=0, delta;
    memset(pre, 0, sizeof(pre));
    for(ll i=1;i<=n;i++) slack[i] = inf;
    matched[y] = u;
    while(1){
        x = matched[y], delta = inf, visy[y] = 1;
        for(ll i=1;i<=n;i++){
            if(visy[i]) continue;
            if(slack[i] > ex[x]+ey[i]-mp[x][i])
                slack[i] = ex[x]+ey[i]-mp[x][i], pre[i] = y;
            if(slack[i] < delta) delta = slack[i], newy = i;
        }
        for(ll i=0;i<=n;i++){
            if(visy[i]) ex[matched[i]] -= delta, ey[i] += delta;
            else slack[i] -= delta;
        }
        y = newy;
        if(matched[y]==-1) break;
    }
    while(y) matched[y] = matched[pre[y]], y=pre[y];
}

```

```

ll km(){
    memset(matched, -1, sizeof(matched));
    memset(ex, 0, sizeof(ex));
    memset(ey, 0, sizeof(ey));
    for(ll i=1;i<=n;i++){
        memset(visy, 0, sizeof(visy));
        match(i);
    }
    ll result = 0;
    for(ll i=1;i<=n;i++)
        if(matched[i] != -1) result += mp[matched[i]][i];
    return result;
}

```

KM算法解决的最大权匹配，最小值问题可以用负数去替代：

```
for(11 i=1;i<=km0.m;i++){
    11 u, v, w;
    //计算距离得出权值w
    km0.add_flow(u, v, -w);
}
```

最后输出负最大权的负，就是权值的最小值：

```
printf("%lld\n", -km());
```

时间复杂度

KM算法时间复杂度为 $O(n^3)$ ，而题目中 n 最大为240，已经足已解决问题。