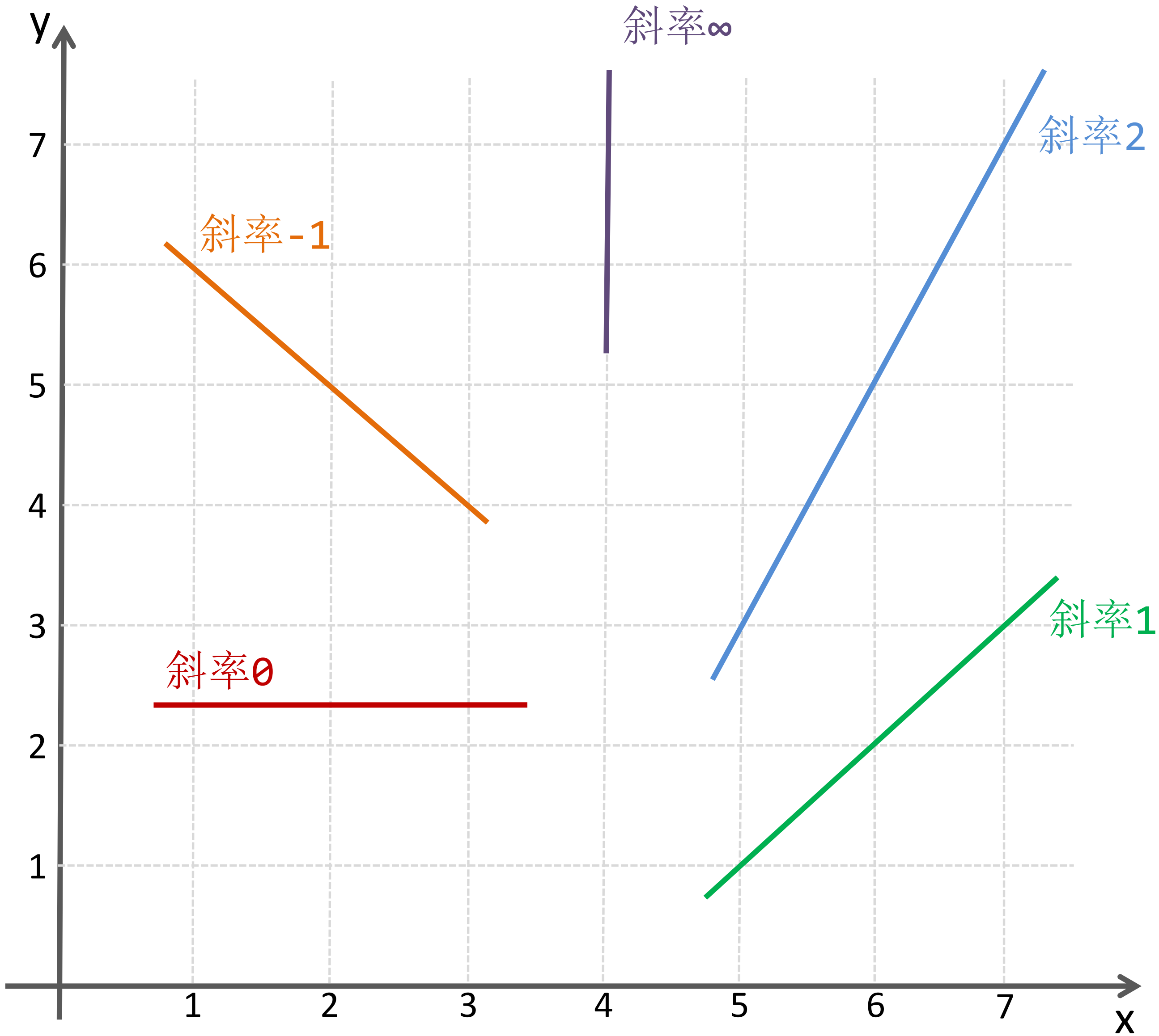
C++算法

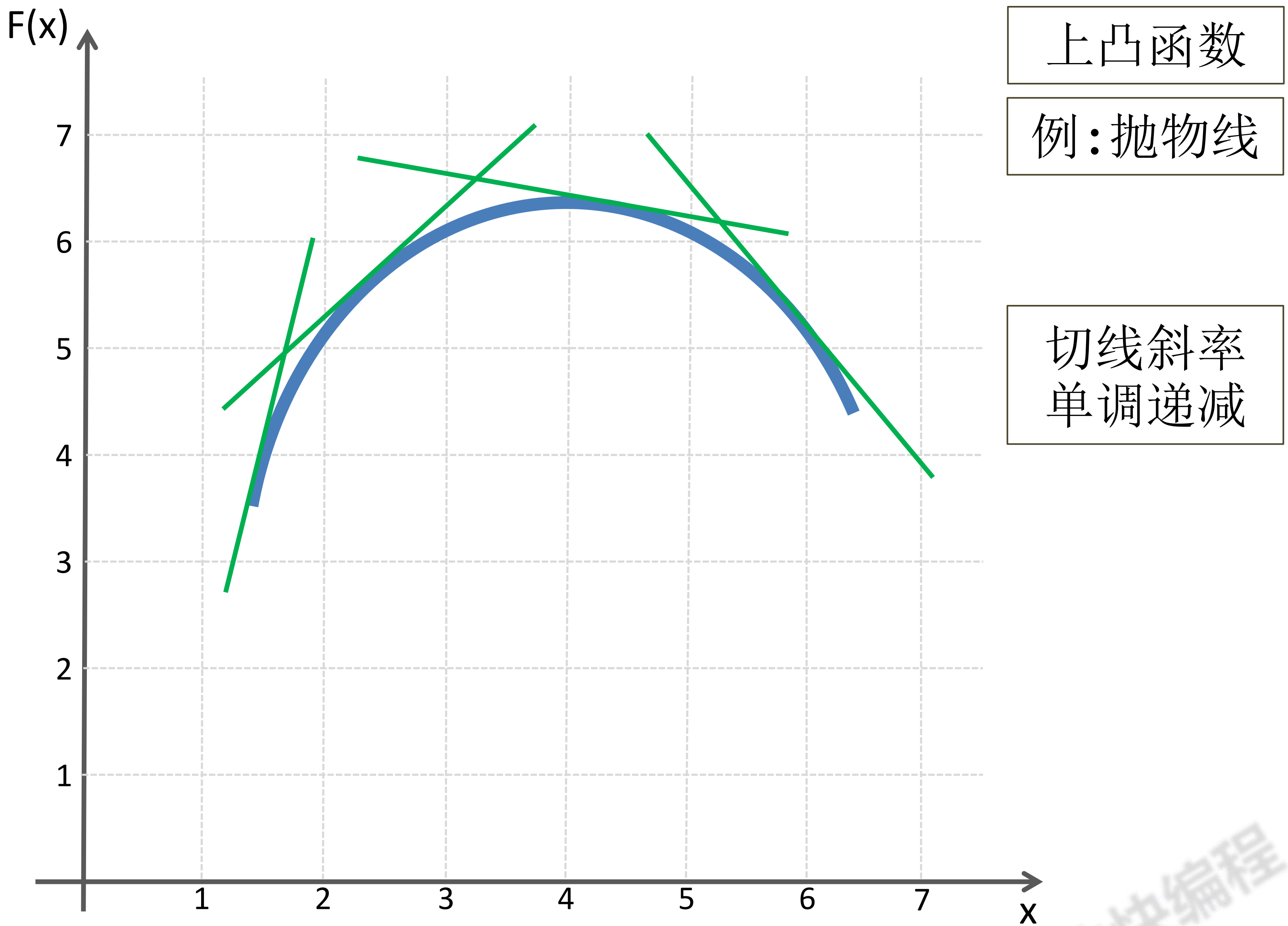
直线斜率

$$\frac{\Delta y}{\Delta x}$$

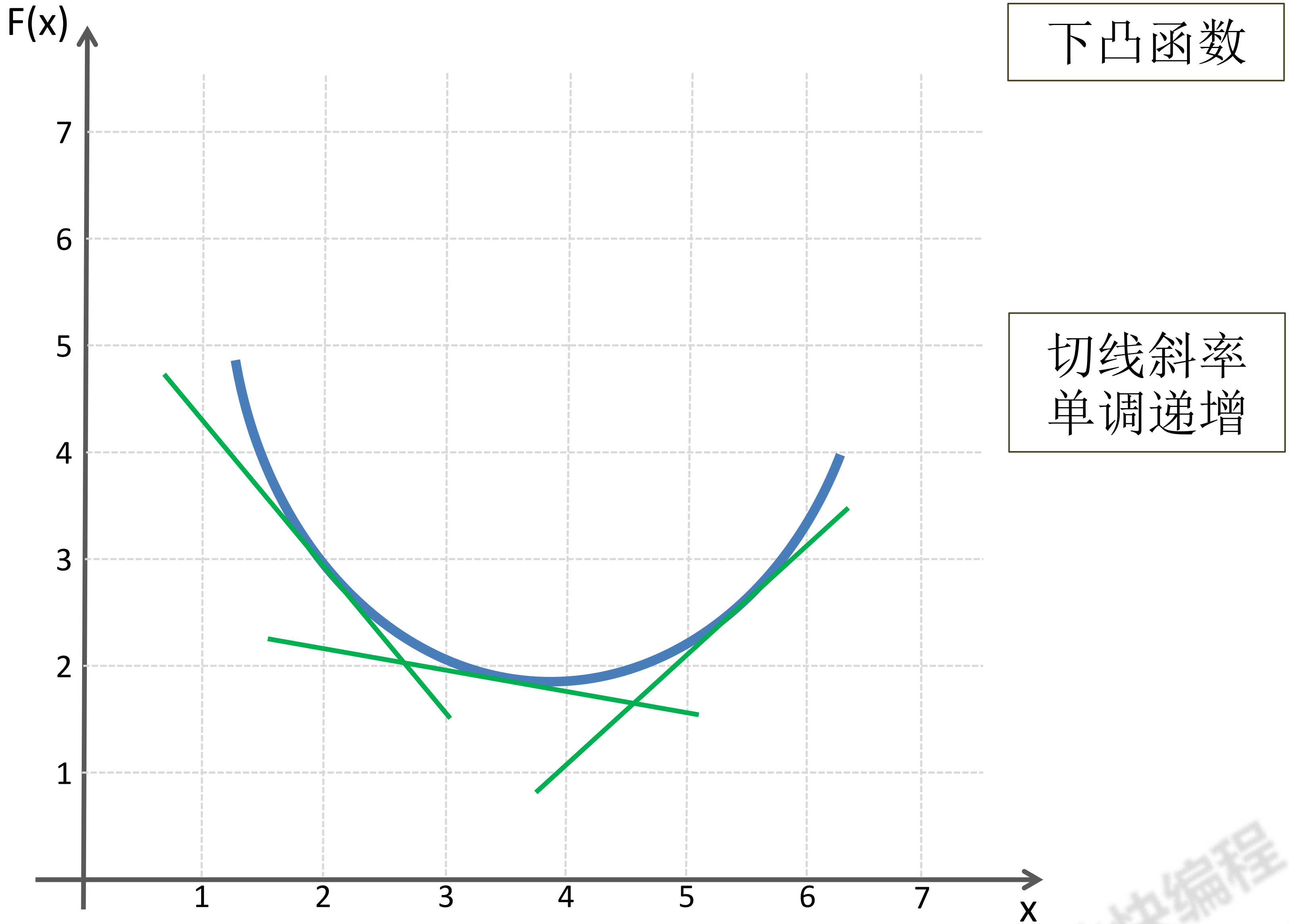


凸函数

convex function

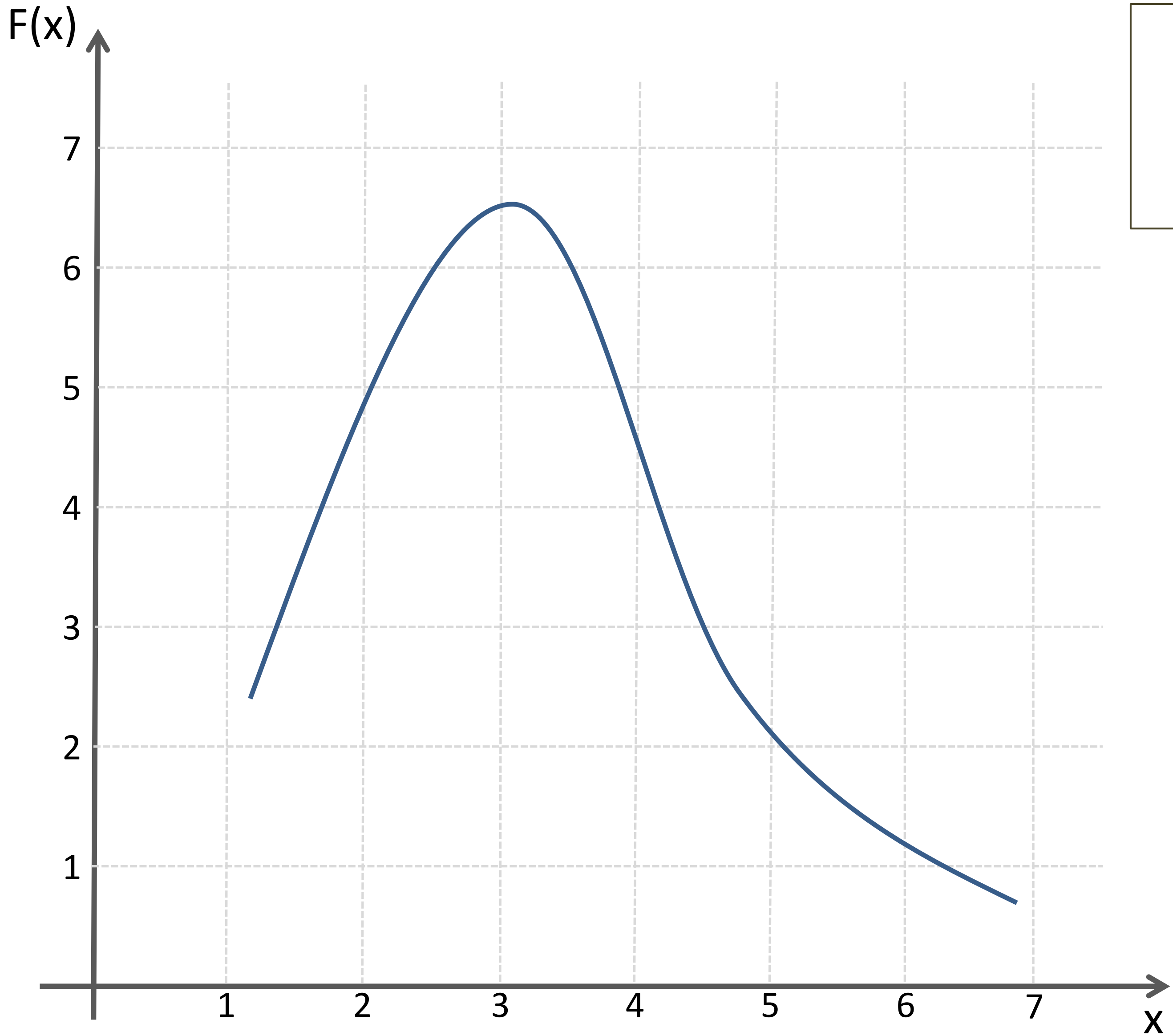


F(x)
7
6
5
4
3
2
1
1
2
3
4
5
6
7
x
上凸函数
例：抛物线
切线斜率
单调递减

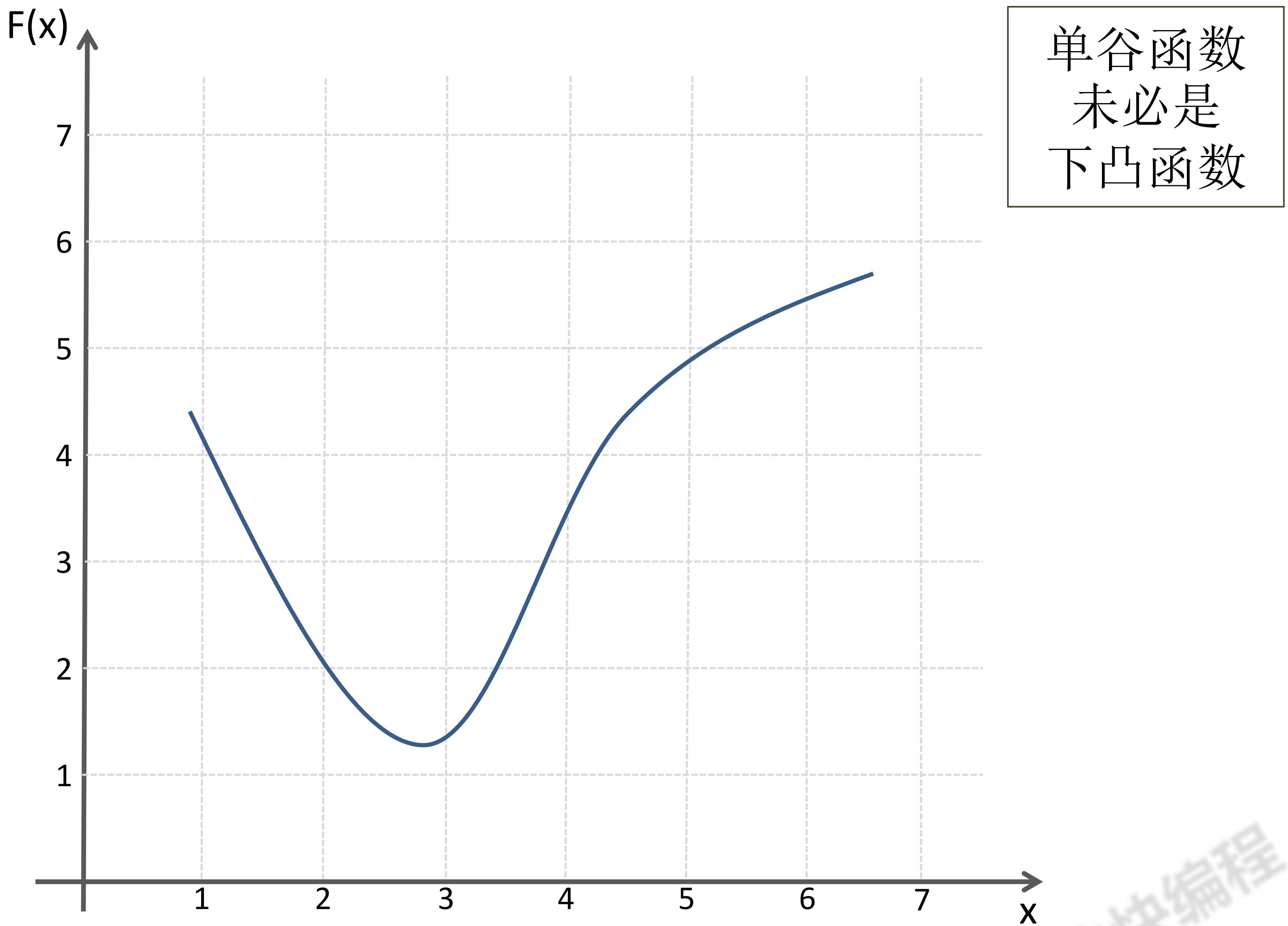


F(x)
7
6
5
4
3
2
1
1
2
3
4
5
6
7
x
下凸函数
切线斜率
单调递增

单峰函数



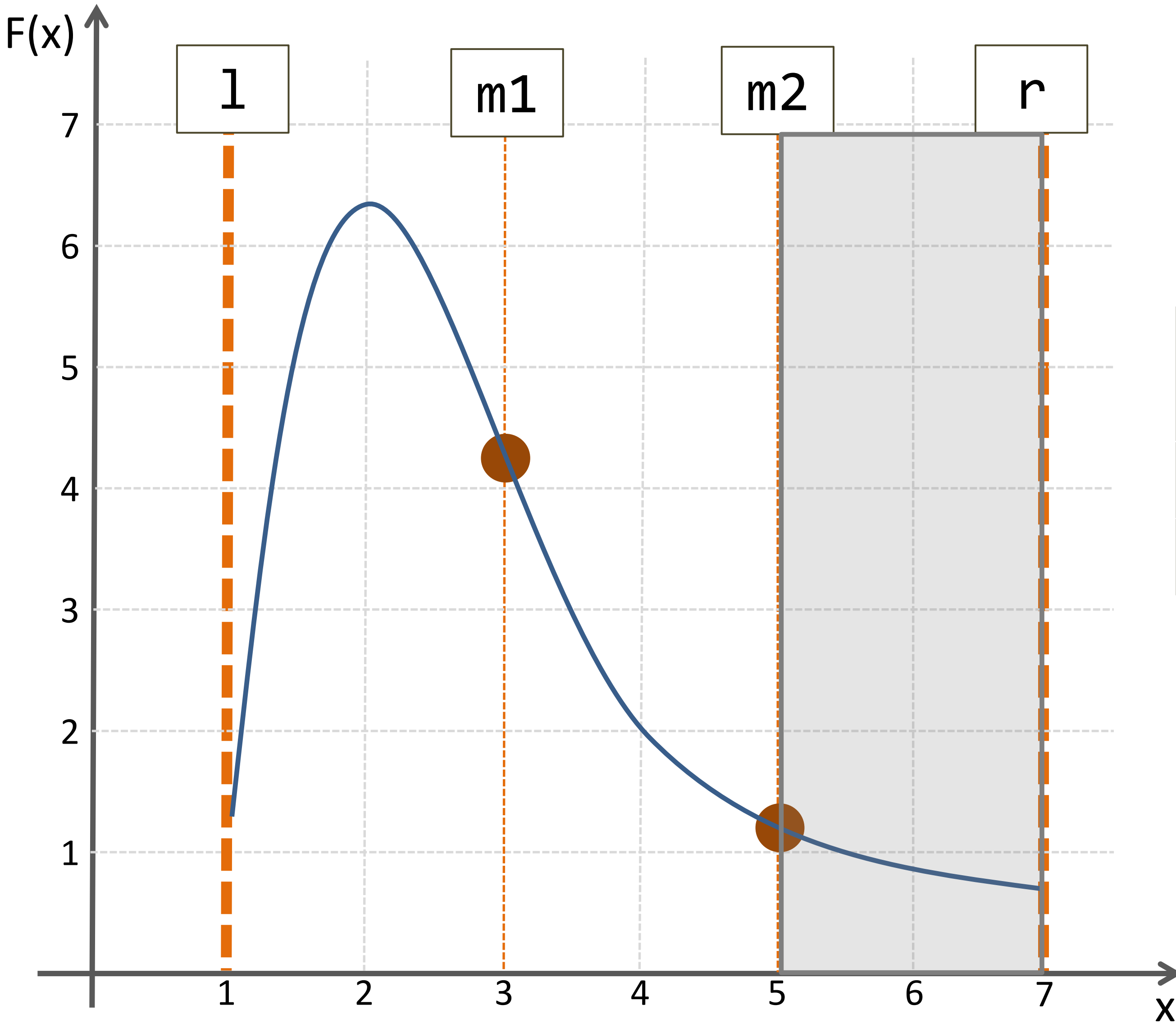
单峰函数
未必是
上凸函数


F(x)
7
6
5
4
3
2
1
1
2
3
4
5
6
7
x
单谷函数
未必是
下凸函数

单峰函数求最值

三分法

x范围[1,7]
求F(x)最大值



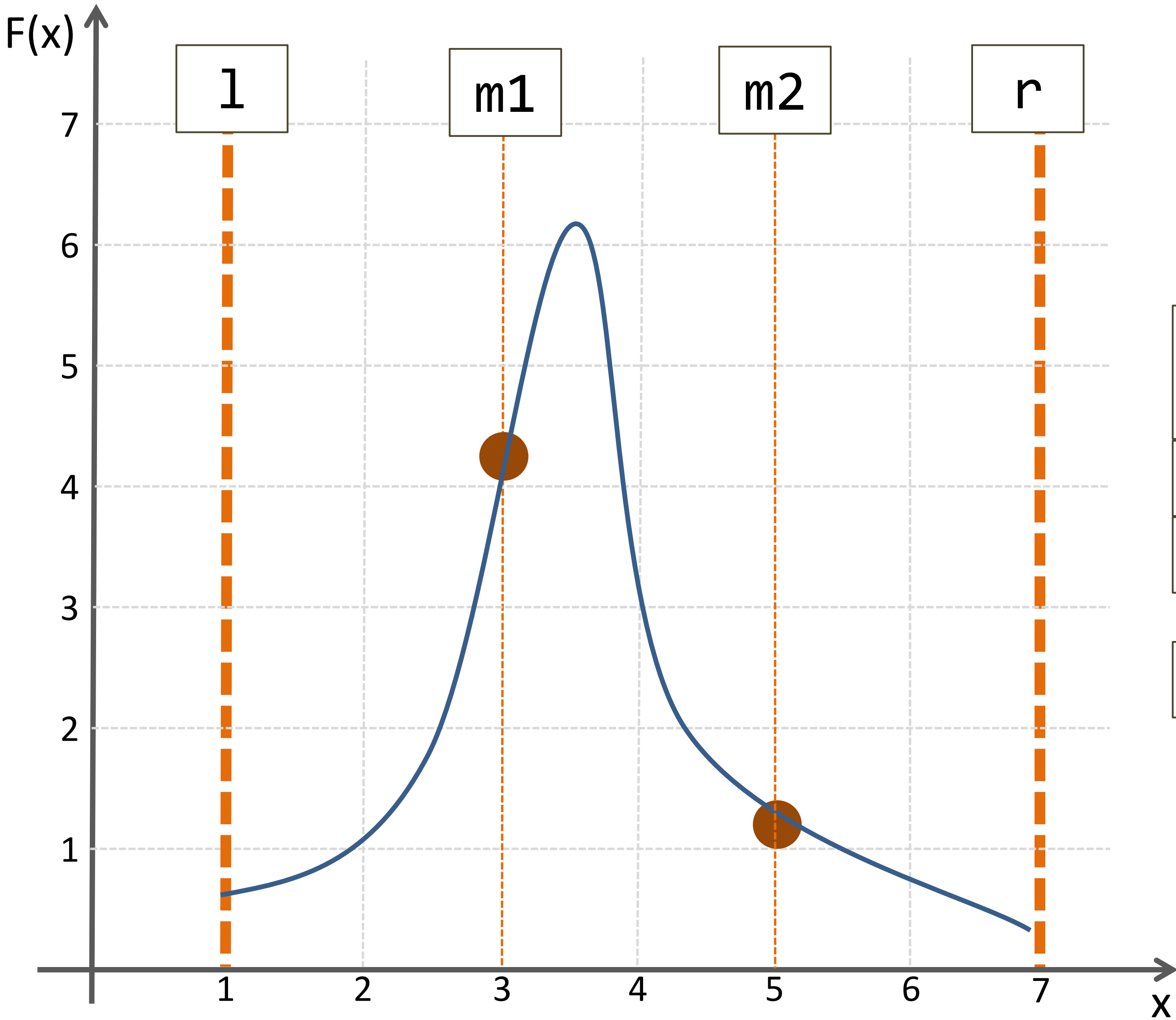
m1,m2将范围划分为
左中右三部分

发现F(m1)>F(m2)

最大值不可能在右侧

三分法

x范围[1,7]
求F(x)最大值



m1,m2将范围划分为
左中右三部分

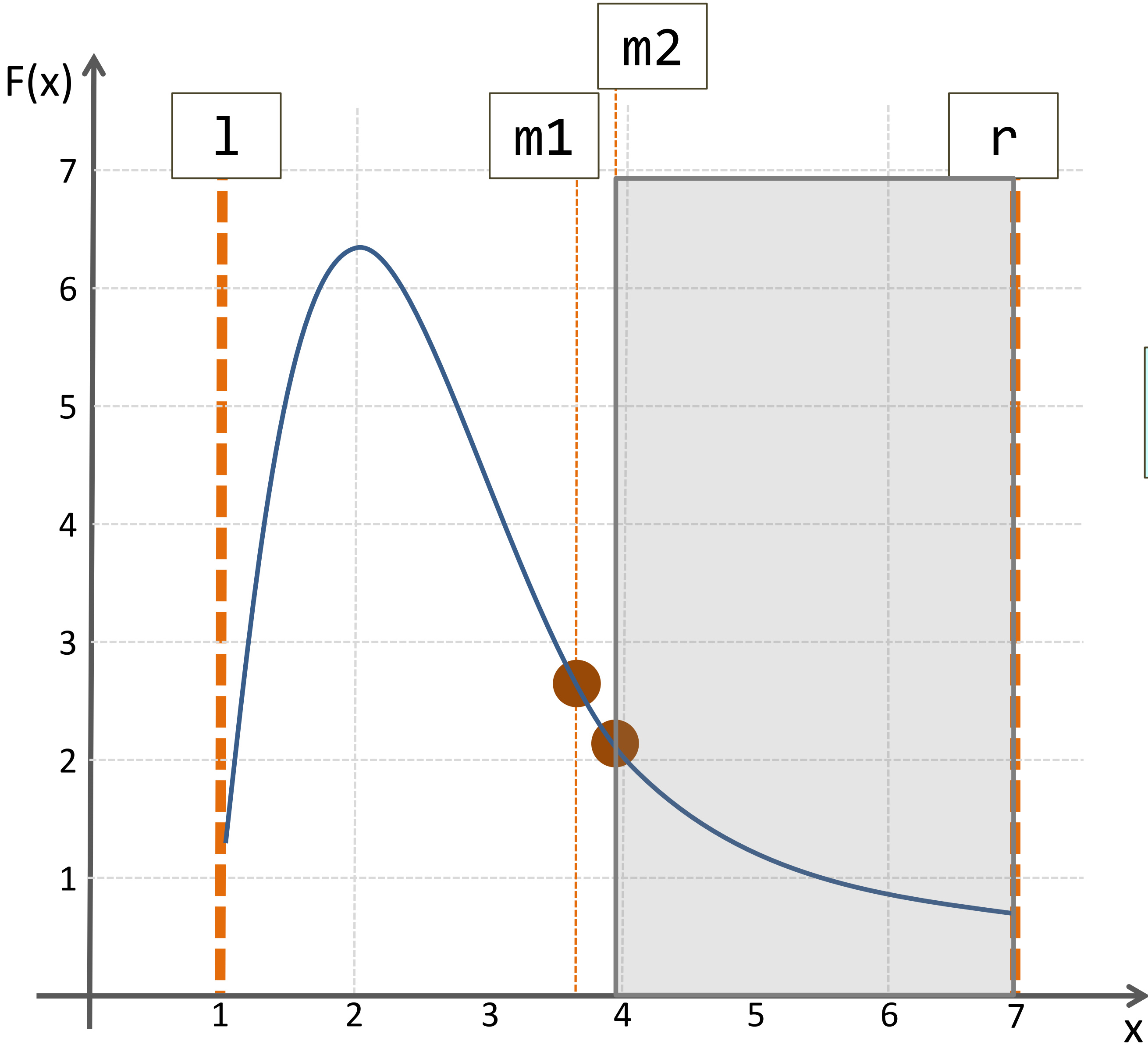
发现F(m1)>F(m2)

最大值不可能在右侧

最大值可能在中间吗?

三分法

x范围[1,7]
求F(x)最大值



m1,m2取在中点附近
每轮排除的范围更多

快快编程2620

$$F(x)= x^7+x^6+x^3+x^2-y*x, \quad 0<y<100$$

函数的二维平面图像并不清楚

暴力枚举

枚举对象	决策变量/自变量 x
枚举范围	[0,100]
	因为$0<y<100$ 所以当$x>100$时 $x^2-y*x=(x-y)x$单调增加 $F(x)$单调增加

打表观察：单谷

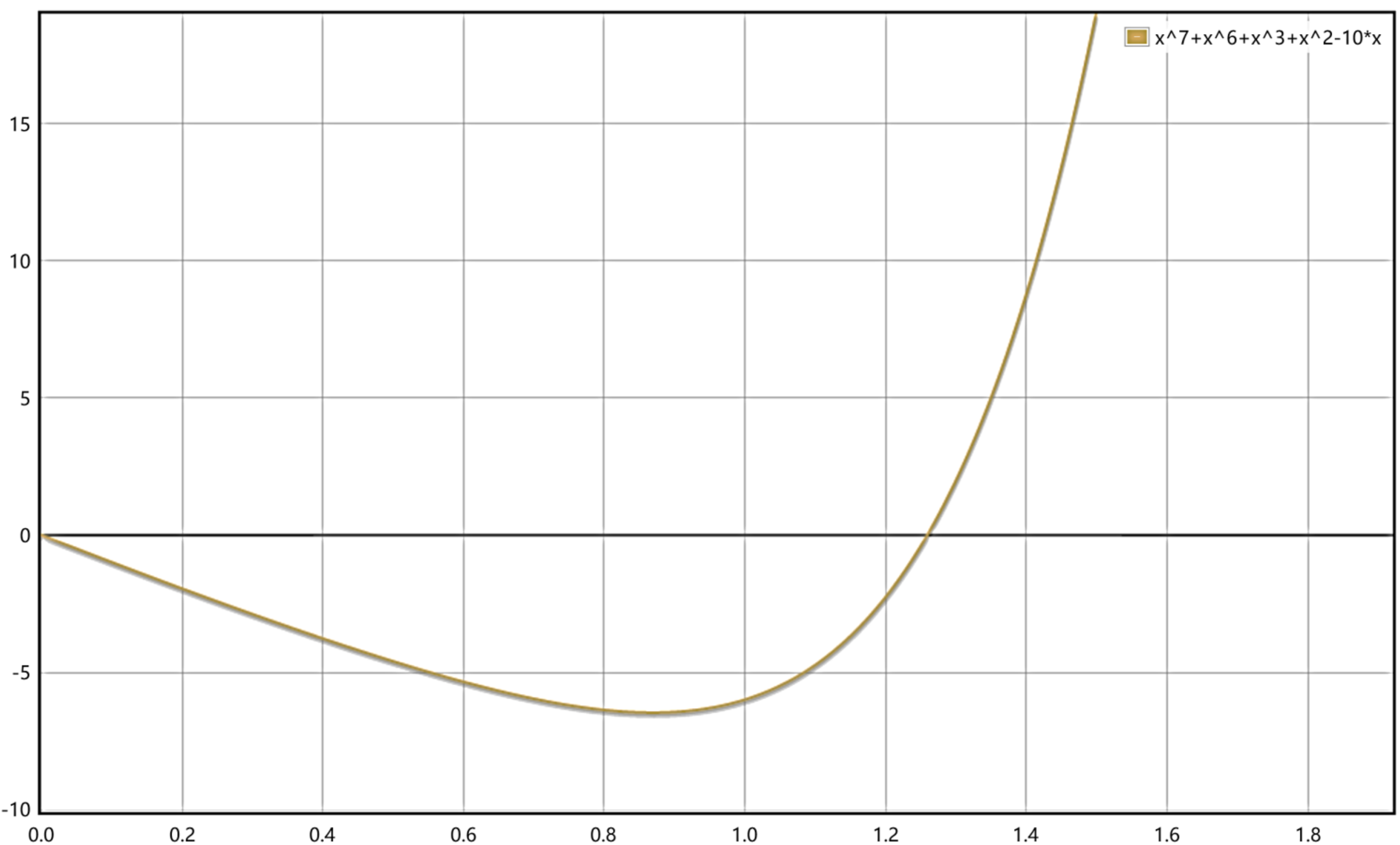
y=10

```cpp
    int T;
    cin>>T;
    for(int i=1;i<=T;++i){
        cin>>y;
        ld minX=0;
        ld minVal=1e18;
        for(ld x=0;x<=100;x+=0.001){
            ld val=F(x);
            if(val<minVal){
                minX=x;
                minVal=val;
            }
            cout<<"F("<<x<<")="<<val<<endl;
        }
        cout<<fixed<<setprecision(3)<<minVal<<endl;
    }
```

```
F(0)=0
F(0.001)=-0.009999
F(0.002)=-0.019996
F(0.003)=-0.029991
F(0.004)=-0.0399839
F(0.005)=-0.0499749
F(0.006)=-0.0599638
F(0.007)=-0.0699507
F(0.008)=-0.0799355
F(0.009)=-0.0899183
F(0.01)=-0.099899
F(0.011)=-0.109878
F(0.012)=-0.119854
F(0.013)=-0.129829
F(0.014)=-0.139801
F(0.015)=-0.149772
F(0.016)=-0.15974
F(0.017)=-0.169706
F(0.018)=-0.17967
F(0.019)=-0.189632
F(0.02)=-0.199592
F(0.021)=-0.20955
F(0.022)=-0.219505
F(0.023)=-0.229459
F(0.024)=-0.23941
F(0.025)=-0.249359
F(0.026)=-0.259306
F(0.027)=-0.269251
F(0.028)=-0.279194
F(0.029)=-0.289135
F(0.03)=-0.299073
F(0.031)=-0.309009
F(0.032)=-0.318943
F(0.033)=-0.328875
F(0.034)=-0.338805
F(0.035)=-0.348732
F(0.036)=-0.358657
F(0.037)=-0.36858
F(0.038)=-0.378501
F(0.039)=-0.38842
F(0.04)=-0.398336
F(0.041)=-0.40825
```

打表观察：单谷

y=10



F(0.855)=-6.46928
F(0.856)=-6.46988
F(0.857)=-6.47043
F(0.858)=-6.47095
F(0.859)=-6.47142
F(0.86)=-6.47185
F(0.861)=-6.47223
F(0.862)=-6.47258
F(0.863)=-6.47287
F(0.864)=-6.47313
F(0.865)=-6.47334
F(0.866)=-6.4735
F(0.867)=-6.47362
F(0.868)=-6.4737
F(0.869)=-6.47373
F(0.87)=-6.47372
F(0.871)=-6.47366
F(0.872)=-6.47355
F(0.873)=-6.4734
F(0.874)=-6.4732
F(0.875)=-6.47296
F(0.876)=-6.47267
F(0.877)=-6.47234
F(0.878)=-6.47195
F(0.879)=-6.47153
F(0.88)=-6.47105
F(0.881)=-6.47052
F(0.882)=-6.46995
F(0.883)=-6.46933
F(0.884)=-6.46866
F(0.885)=-6.46795
F(0.886)=-6.46718
F(0.887)=-6.46637
F(0.888)=-6.46551

三分法

浮点数框架

```
typedef long double ld;
const ld ERR=0.000001;
const ld DELTA=ERR/10;
```

```
        ld l=0;
        ld r=100;
        while(r-l>ERR){
            ld ml=(l+r)/2;
            ld mr=ml+DELTA;
            if(F(ml)<F(mr))
                r=mr;
            else
                l=ml;
        }
        ld ans=F(l);
```

快快编程2621

贪心法

优先使用没用过的干净桌布	费用为0
再使用便宜的清洗方案	费用为c1
最后使用贵的清洗方案	费用为c2

```
if(c1>c2){swap(n1,n2);swap(c1,c2);}
if(n1<=n2){c2=c1;n2=n1;}
```

贪心法

变量m表示购买的干净桌布还剩几张没用过

容器q1维护前n1天及以前使用过
但还没定清洗方案的桌布信息

容器q2维护前n2天到前n1-1天使用过
但还没定清洗方案的桌布信息

双端
队列

```
struct Cover{int date,num;};
deque<Cover> q1,q2;
```

	i=1	i=2	i=3	i=4	i=5
d[i]=	5	3	2	6	10

T=5	共5天
n1=3	清洗方案1：3天洗完
n2=1	清洗方案2：1天洗完
c1=10	清洗方案1：每张桌布10元
c2=20	清洗方案2：每张桌布20元
m=10	初始时共10张桌布

容器调配

```
struct Cover{int date,num;};
int F(int m){
    int cost=0;
    deque<Cover> q1,q2;
    for(int i=1;i<=T;++i){
        int demand=d[i];
        int cnt=min(demand,m);
        demand-=cnt;
        m-=cnt;
        if(i-n2>0) q2.push_back((Cover){i-n2,d[i-n2]});
        q2容器里较早桌布导入q1
        q1容器里较早桌布确定清洗方案1
        q2容器里较晚桌布确定清洗方案2
        if(demand) return -1;
    }
    return cost;
}
```

快快编程2622

请同学写出题目大意

识别核心决策

m表示购买几张桌布

总费用F(m)=g(m)+m*p

g(m)表示清洗费用

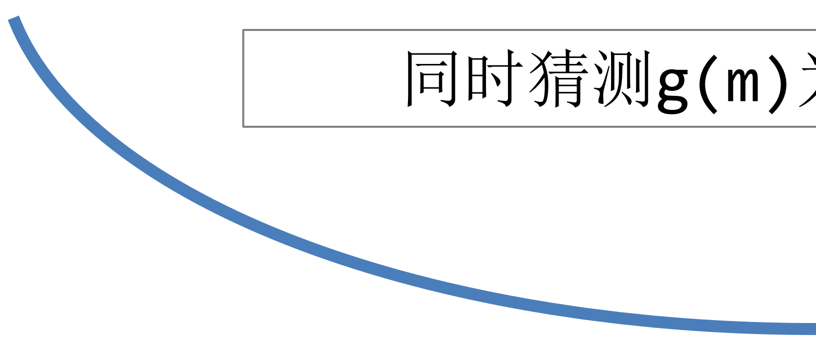
判断函数凸性

m较小时无解，g(m)为INF

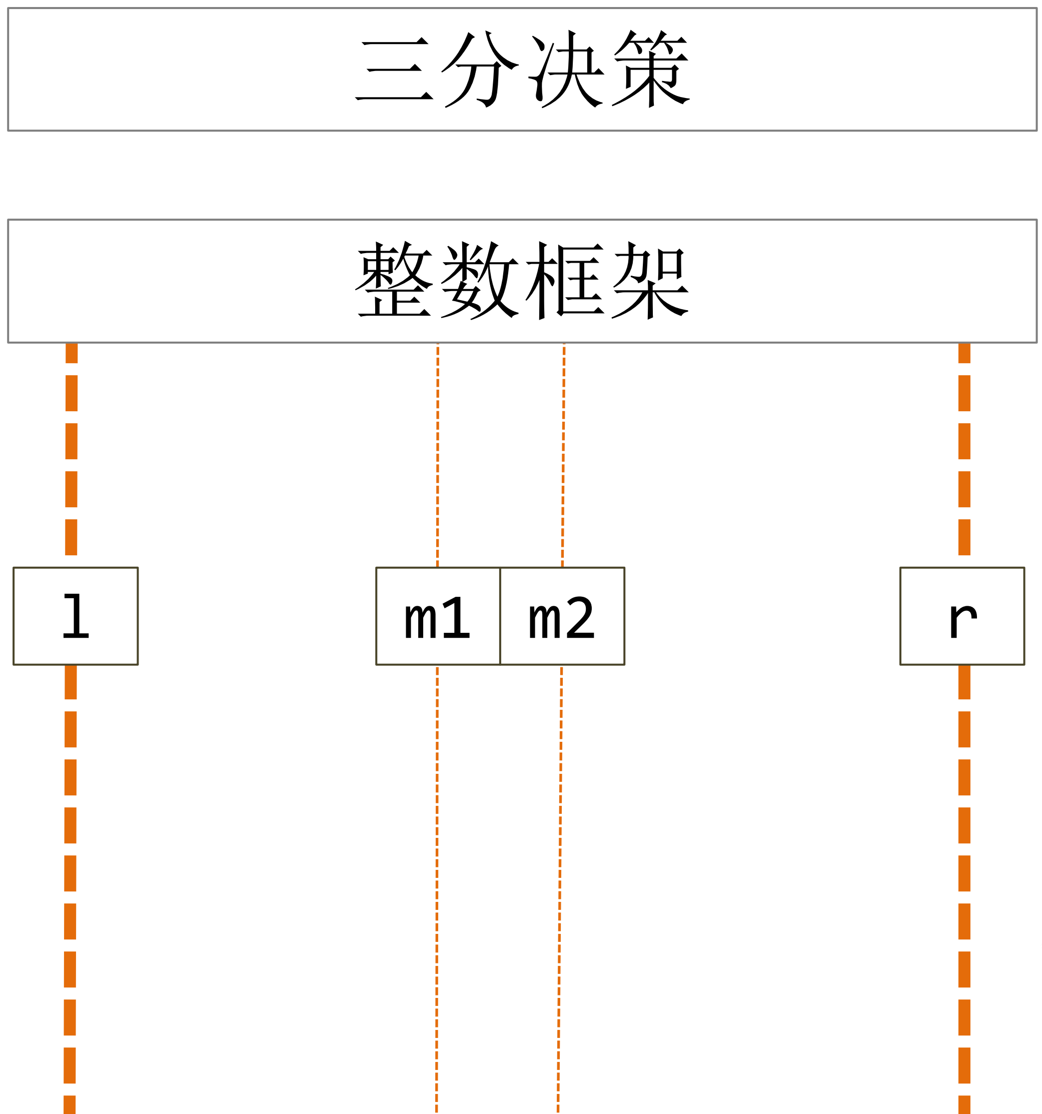
m较大时g(m)维持不变

所以猜测g(m)单调下降

同时猜测g(m)为下凸函数



```
int l=0;
int r=T*50;
int ans=r*p;
for(int i=l;i<=r;++i)ans=min(ans,F(i));
```


三分决策
整数框架
l
m1
m2
r

三分决策

整数框架

```
int l=0;
int r=T*50;
int ans=r*p;
while(r-l>=3){
    int ml=l+(r-l)/2;
    int mr=ml+1;
    int Fml=F(ml);
    if(Fml==INF){l=ml+1;continue;}
    int Fmr=F(mr);
    if(Fml<Fmr){ans=Fml;r=mr-1;}
    else{ans=Fmr;l=ml+1;}
}
for(int i=l;i<=r;++i)ans=min(ans,F(i));
```

3能否改成10?

```
    int l=0;
    int r=T*50;
    int ans=r*p;
//  while(r-l>=3){
//      int ml=l+(r-l)/2;
//      int mr=ml+1;
//      int Fml=F(ml);
//      if(Fml==INF){l=ml+1;continue;}
//      int Fmr=F(mr);
//      if(Fml<Fmr){ans=Fml;r=mr-1;}
//      else{ans=Fmr;l=ml+1;}
//  }
    for(int i=l;i<=r;++i)ans=min(ans,F(i));
```

三分法	枚举决策 横坐标	比较答案大小
二分法	枚举答案 纵坐标	判断可行性

快快编程

2620,2621,2622