

题意解释

有一个包含(,), ? 的序列, 将 ? 替换成 (或者), 使得不能被匹配的括号数目尽量少。

知识点提炼

贪心, 栈, 动态规划

核心解题思路

思路一: $n \leq 20$

爆搜, $O(2^n)$ 。期望得分 20 分。

思路二: $n \leq 100$

记 $f_{i,j,k}$ 表示长度为 i 的前缀当前有 j 个还能用的左括号, 当前答案是 k 是否可行, 做一个可行性DP, 构造方案直接还原就行, 复杂度 $O(n^3)$ 。期望得分 40 分。

思路三: $n \leq 1000$

因为是让答案最小, 所以直接 $f_{i,j}$ 表示长度为 i 的前缀有 j 个还能用的左括号的最小答案, 做最优化DP, 构造方案类似, 复杂度 $O(n^2)$ 。期望得分 70 分。

思路四: $n \leq 1e5$

实际上直接用栈维护贪心就好。

开两个栈分别维护前缀能用的左括号和问号, 遇到右括号时优先用左括号, 否则用问号, 否则不能被匹配的括号数加一。用问号时左括号一定是空的, 所以使用的顺序不会影响后面的贪心。

之后考虑匹配剩余的左括号和问号。从左到右, 找到每一个左括号右侧第一个问号匹配掉即可, 否则不能被匹配的括号数加一。

最后考虑匹配剩下的问号。两两匹配即可, 如果剩余问号数是奇数, 不能被匹配的括号数加一。

容易证明这个策略是足够优秀的, 复杂度 $O(n)$ 。构造直接在贪心过程中就能完成。

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef double dou;
typedef pair<int,int> pii;
#define fi first
#define se second
#define mapa make_pair
typedef long double ld;
typedef unsigned long long ull;
template <typename T>inline void read(T &x){
    x=0;char c=getchar();bool f=0;
    for(;c<'0' || c>'9';c=getchar()) f|=(c=='-');
    for(;c>='0'&&c<='9';c=getchar())
        x=(x<<1)+(x<<3)+(c^48);
```

```

    x=(f?-x:x);
}
const int N=1e5+5, mod=998244353;
int T, n, m;
char s[N];
int s1[N], s2[N], t1, t2;
int main(){
    read(T);
    while(T--){
        read(n);
        scanf("%s", s+1);
        t1=t2=0;
        int ans=0;
        for(int i=1; i<=n; ++i){
            if(s[i]=='('){
                s1[++t1]=i; // s1 : 左括号的栈
            }
            else if(s[i]==')'){
                if(t1) --t1;
                else if(t2){
                    s[s2[t2]]='(';
                    --t2;
                }
                else{
                    ++ans; // 有左取左，无左取问，否则答案+1
                }
            }
            else{
                s2[++t2]=i; // s2 : 问号的栈
            }
        }
        while(t1){
            if(t2&& s2[t2]>s1[t1]){
                s[s2[t2]]=')'; --t2;
                --t1;
            }
            else --t1, ++ans; // 每个左括号找右边的问号
        }
        ans+=t2&1;
        for(int i=1; i<=t2; ++i){ // 剩余的问号两两匹配
            if(i&1) s[s2[i]]='(';
            else s[s2[i]]=')';
        }
        printf("%d\n", ans);
        for(int i=1; i<=n; ++i) putchar(s[i]);
        putchar('\n');
    }

    return 0;
}

```

本题易错点

- 设计贪心策略时，容易遗漏一些分类讨论