BEAULC Solution:

Trước tiên ta xét bài toán đơn giản hơn khi chỉ có 1 dãy ngoặc $s_1s_2 \dots s_n$. Hãy đếm số dãy con là biểu thức ngoặc đúng.

Lập mảng $d_0 = 0, d_1, d_2, \dots, d_n$ với $d_i = d_{i-1} + 1$ khi $s_i = '('$ và $d_i = d_{i-1} - 1$ nếu $s_i = ')'$. Khi đó một đoạn $s_{i+1} \dots s_j$ là một dãy ngoặc đúng khi:

- $d_i = d_j$
- $d_k \geq d_j \ \forall k = i, \dots, j$

Gọi Pr[i]=k là chỉ số nhỏ nhất (k<i) sao cho d[u]≥d[i] với mọi u=k,...,i (BT cơ bản)

Gọi L[i] là vị trí gần nhất về bên trái mà d[L[i]]=d[i]

Đặt f[i] là số lượng biểu thức ngoặc đúng kết thúc tại i ta có:

$$f[i] = \begin{cases} f\left[L[i]\right] + 1 \text{ nếu } \mathrm{L[i]} \geq \mathrm{Pr[i]} \\ 0 \text{ nếu } L[i] < \mathrm{Pr}[i] \end{cases}$$

Tổng các giá trị f[i] là đáp số.

Mở rộng ra k dãy:

Lập k mảng đếm (dấu '(' cộng 1 còn dấu ')' thì trừ 1:

$$d_{1,0} = 0, d_{1,1}, d_{1,2}, \dots, d_{1,n}$$
$$d_{2,0} = 0, d_{2,1}, d_{2,2} \dots, d_{2,n}$$
$$\dots \dots \dots \dots \dots \dots \dots \dots .$$
$$d_{k,0} = 0, d_{k,1}, d_{k,2}, \dots ., d_{k,n}$$

Lưu kết quả vào mảng vector:

vector<int> d[maxn];

(mỗi phần tử của d là vector k phần tử)

B2: Với mỗi dãy u $(u = 1 \div k)$ tại vị trí i tính giá trị Pr[u,i] = chỉ số nhỏ nhất trước i mà từ chỉ số này đến i tất cả các giá trị $d_{u,?} \geq d_{u,i}$ và Pmax[i] là giá trị lớn nhất trong số k giá trị này.

B3: Với mỗi dãy u $(u = 1,2, \dots, k)$ tại vị trí i tính L[u,i]=chỉ số lớn nhất mà vector $(d_{1,i} \dots, d_{k,i})$ xuất hiện trước i

```
map<vector<int>,int> nho;
nho[d[0]]=0;
for(int i=1;i≤n;++i) {
   if (nho.find(d[i])!=nho.end()) L[i]=nho[d[i]];
   else L[i]=-1;
   nho[d[i]]=i;
}
```

B4: Đặt f[i] là số k biểu thức ngoặc đồng thời kết thúc tại i. Ta có công thức qui hoạch động:

- f[i]=f[L[i]]+1 nếu L[i]≥Pmax[i]
- f[i]=0 trong trường hợp ngược lại

Kết quả là tổng các phần tử của mảng f