پاسخ سوال اول:

الگوریتم پیچیده:

```
int sum = 0
for(int i=1; i<n; i*=2)
    for(int j=n; j>0; j/=2)
        for(int k=j; k<n; k+=2)
            sum += i*j*k
return sum
```

- `for(int i=1; i<n; i*=2)` → B_1 را $\lg n$ بار اجرا کند
- B_1: `for(int j=n; j>0; j/=2)` → B_2 را $\lg n$ بار اجرا کند
- B_2: `for(int k=j; k<n; k+=2)` → $\boxed{B_3 \text{ را } \frac{n-j}{2} \text{ بار اجرا کند}}$ → باید j از رابطه حذف شود
- B_3: `sum += i*j*k`

⇓

$\boxed{n, n/2, n/4, n/8, \dots, 1}$ ← باید مقادیر j مشخص شود

⇓

بنابراین با هر بار اجرای B_2، تعداد اجرای B_3 تغییر کند. پس باید به دنبال جمع اجرای B_3 برای اجراهای مختلف B_2 باشیم:

در اجرای k ام از B_2 خواهیم داشت:

$$\text{تعداد اجرای } B_3 = \left(n - n/2^{k-1}\right)/2$$

$$\Longrightarrow \underbrace{\text{تعداد اجرای } B_3 \text{ در یکبار اجرای } B_1}_{A} = \sum_{k=1}^{\lg n} \left(n - n/2^{k-1}\right)/2$$

$$A = \frac{1}{2}\left[\underbrace{\left(\sum_{k=1}^{\lg n} n\right)}_{n\lg n} - \left(n \underbrace{\sum_{k=1}^{\lg n} \frac{1}{2^{k-1}}}_{C}\right)\right] = \frac{1}{2}(n\lg n - nc) = \boxed{\frac{1}{2} n(\lg n - c)}$$

اگر k به بینهایت میل کند حاصل با فرمول $\frac{a}{1-q}$ بدست می آید. بنابراین این مقدار همواره کوچکتر از $\frac{1}{1-\frac{1}{2}}$ یعنی ۲ خواهد بود و از طرفی بزرگتر از ۱ است

⇓

پس c بین ۱ و ۲ است

⇓

$\boxed{1 < c < 2}$

تعداد اجرای B_3 در کل برنامه $\Leftarrow T = A \lg n$

$T = \frac{1}{2} n \lg n (\lg n - c), \quad 1 < c < 2$

$\boxed{T \approx O(n \lg^2 n)}$

← با توجه به اینکه تعداد عملیات اصلی در این سوال برابر با تعداد اجرای B_3 است، پس این تابع دارای پیچیدگی زمانی $O(n\lg^2 n)$ است.

رابطه پیچیده:

$$T(n) = T(n-1) + 3n$$

$$\Rightarrow T(n) = T(n-2) + 3(n-1) + 3n = T(n-3) + 3(n-2) + 3(n-1) + 3n = \underbrace{T(1)}_{T(0)+3\times 1} + 3\times 2 + \cdots + 3\left[(n-2) + (n-1) + n\right]$$

اگر فرض کنیم $T(0) = K$:

$$\Rightarrow T(n) = 1 + 3\left[1 + 2 + \cdots + (n-1) + n\right]$$

$$\Rightarrow T(n) = K + 3\left[\frac{n(n+1)}{2}\right] = \frac{3}{2}n^2 + \frac{3}{2}n + k$$

$$\Rightarrow f(n) = \Theta(n^2)$$

مرتبه پیچیده:

ابتدا دو متغیر min و max می سازیم . سپس آرایه n عضوی را به صورت n/2 جفت در نظر می گیریم. برای هر جفت یک مقایسه برای پیدا کردن کوچکتر بزرگتر آن جفت انجام می دهیم . اگر عدد بزرگتر از max بیش تر بود ، max را آپدیت می کنیم و اگر عدد کوچکتر از min کم تر بود ، min را آپدیت می کنیم . بنابراین در کل به 3n/2 مقایسه نیاز داریم .

به طور دقیق تر اگر n فرد باشد به 3(n-1)/2 و اگر n زوج باشد به 3(n-2)/2 + 1 مقایسه احتیاج داریم . (چون اگر n فرد باشد min و max را با عنصر اول آرایه مقدار دهی اولیه می کنیم و اگر زوج باشد با مینیمم و ماکسیمم جفت اول مقداردهی می کنیم)

پس در کل به کمتر از 3n/2 مقایسه احتیاج داریم .