الگوریتم پیچیده:

پاسخ سوال اول :

$0 < i \le 2^k < n$

$\Rightarrow k < \log_2 n$

k تعداد مراحل اجرای حلقه اول

حلقه اول : $\log n$

$1 \le j = \frac{n}{2^{k'}} \le n$

k' تعداد مراحل اجرای حلقه دوم

حلقه دوم : $\frac{n}{2^{k'}} = 1 \Rightarrow n = 2^{k'} \Rightarrow k' = \log(n)$ آخرین تکرار

حلقه سوم : $j = n, \frac{n}{2}, \frac{n}{4}, \frac{n}{8}, \ldots, 1$

تکرار حلقه سوم = $\frac{n-j}{2} = 0, \frac{n-\frac{n}{2}}{2} = \frac{n}{4}, \frac{3n}{8}, \frac{7n}{16}, \ldots$

پس حلقه سوم به فرم $O(n)$ می باشد.

بنابراین پیچیدگی زمانی الگوریتم به طور کلی $\boxed{O(n\log^2(n))}$ می باشد

رابطه پیچیده:

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

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

اگر فرض کنیم $T(0) = K$:

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

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

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

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 1 + 3(n-2)/2 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 .