بسم الله الرحمن الرحیم

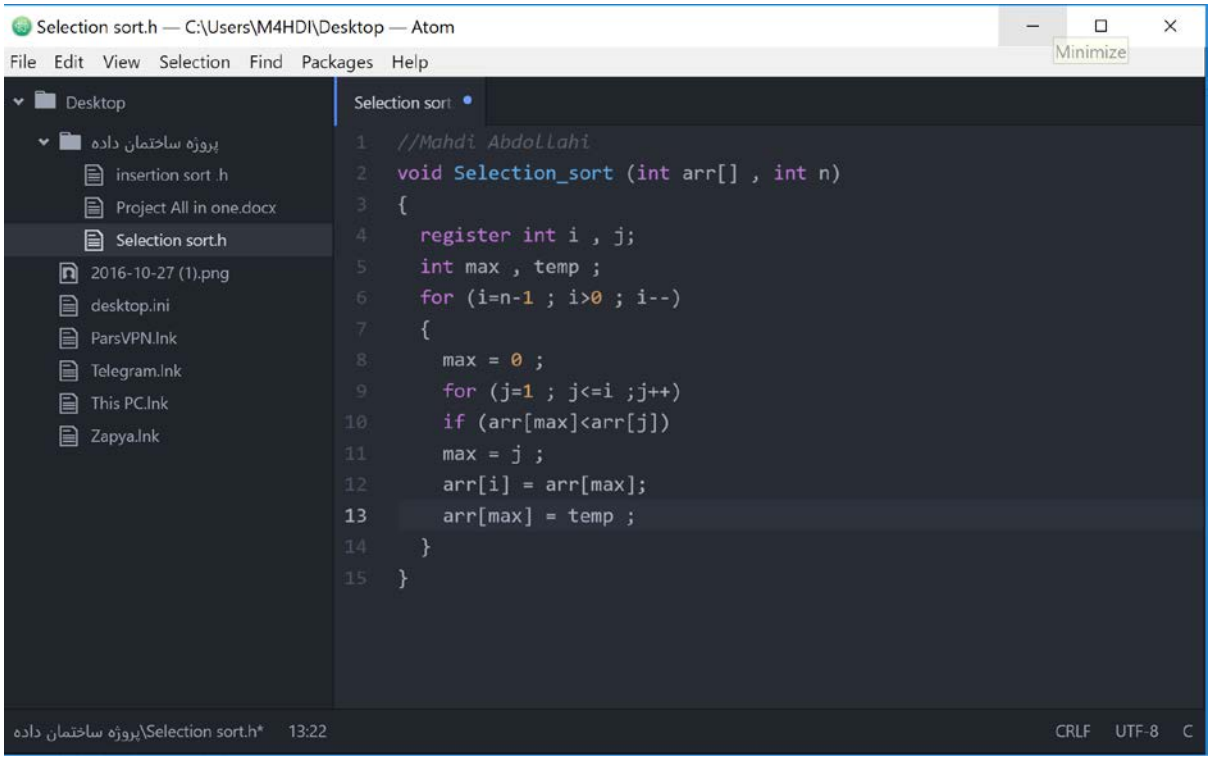
پروژه" Sort " ساختمان داده

نام و نام خانوادگی : مهدی عبدالهی

استاد : جناب آقای هومن سیاری

مرتب سازی انتخابی (Selection_Sort) :

- الگوریتم:



- توضیح فارسی الگوریتم:

در این نوع مرتب سازی الگوریتم بزرگ ترین عدد را از بین بقیه اعداد پیدا کرده و به انتهای لیست انتقال می دهد سپس بقیه اعداد را با هم مقایسه کرده و بزرگترین عدد را کنار عدد انتخابی قبل در انتهای لیست قرار می دهد.

- مثال عددی:

۲۰- ۱۵- ۱۳ – ۳۵ – ۵

ابتدا عدد ۳۵ را پیدا کرده و درانتهای لیست قرار می دهد.

۲۰ - ۱۵ – ۱۳ – ۵ – ۳۵

سپس از ما بقی خانه ها بزرگترین عدد را انتخاب کرده و در کنار آخرین خانه قرار می دهد.

۵ – ۱۵ – ۱۳ - ۲۰ – ۳۵

همین عمل تکرار می شود تا در انتهای اعداد به ترتیب در کنار یکدیگر قرار بگیرند.

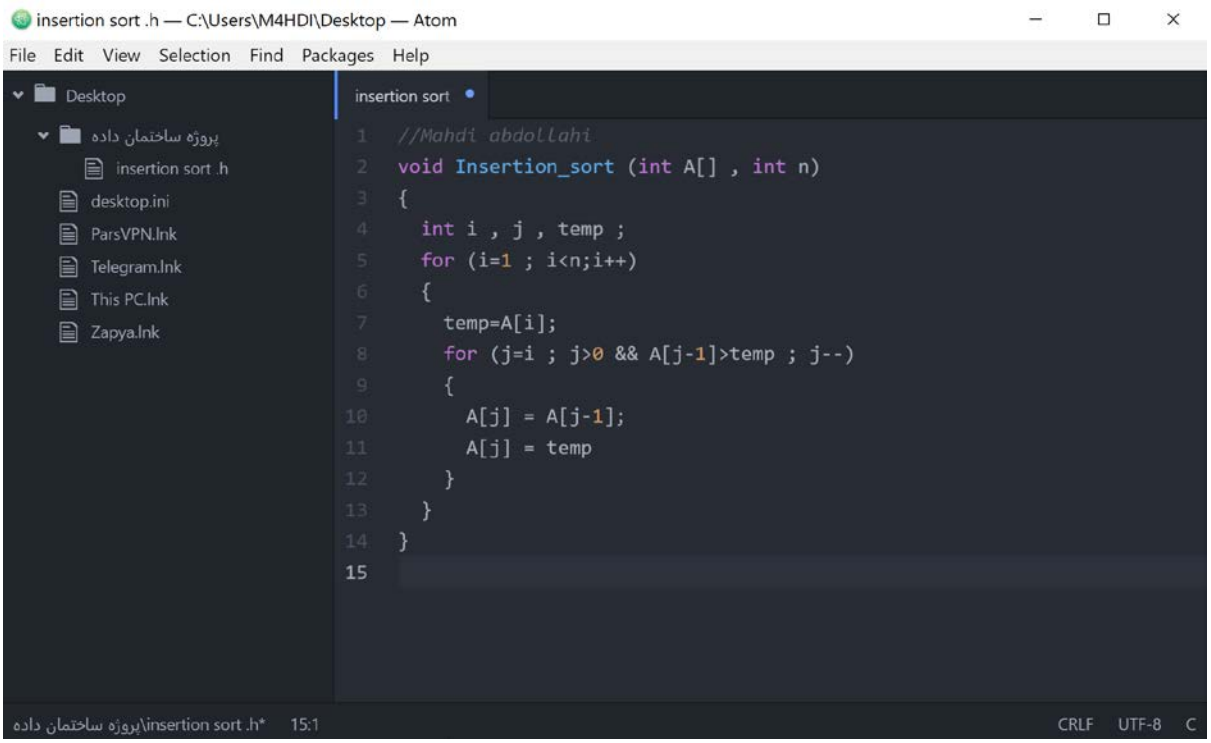
۵ – ۱۳ – ۱۵ – ۲۰ – ۳۵

۵ - ۱۳ – ۱۵ – ۲۰ – ۳۵

- $Big\_O = O(n) = n^2$

مرتب سازی درجی (Insertion_Sort) :

- الگوریتم:



- توضیح فارسی الگوریتم :

در مرتب سازی درجی ابتدا الگوریتم خانه اول و دوم آرایه را با هم مقایسه می کند. پس از مقایسه اعداد را در جای خود قرار می گیرند و یک لیست دوتایی مرتب را تشکیل می دهند سپس خانه سوم آرایه را با خانه اول و دوم مقایسه می کند و اعداد داخل ان را در جای مناسب خود قرار می دهد به طوری که سه خانه آرایه یک لیست مرت سه تایی تشکیل می دهد. الگوریتم به همین صورت پیش می رود تا خانه n ام آرایه را با خانه n-1ا خانه قبل خود مقایسه کند و یک لیست مرتب با n خانه کنار هم ایجاد کند.

- مثال عددی:

۹ – ۵ – ۲۰ – ۱۵ – ۱۳

ابتدا خانه اول و دوم با یکدیگر مقایسه می شود در نتیجه ۵ کوچکتر از ۹ بوده پس جا به چا می شود.

۹ – ۵

۵ – ۹ – ۲۰ – ۱۵ – ۱۳

سپس خانه سوم آرایه مقایسه می شود در این مرحله اعداد در جای خود قرار دارد هیچ تغییری صورت نمی پذیرد.

۵ – ۹ – ۲۰

۵- ۹ – ۲۰ – ۱۵ – ۱۳

در این مرحله خانه چهارم با سه خانه قبل مقایسه می شود و سپس خانه چهارم در جای مناسب خود قرار می گیرد.

۵- ۹ – ۲۰ – ۱۵

۵ – ۹ – ۱۵ – ۲۰ – ۱۳

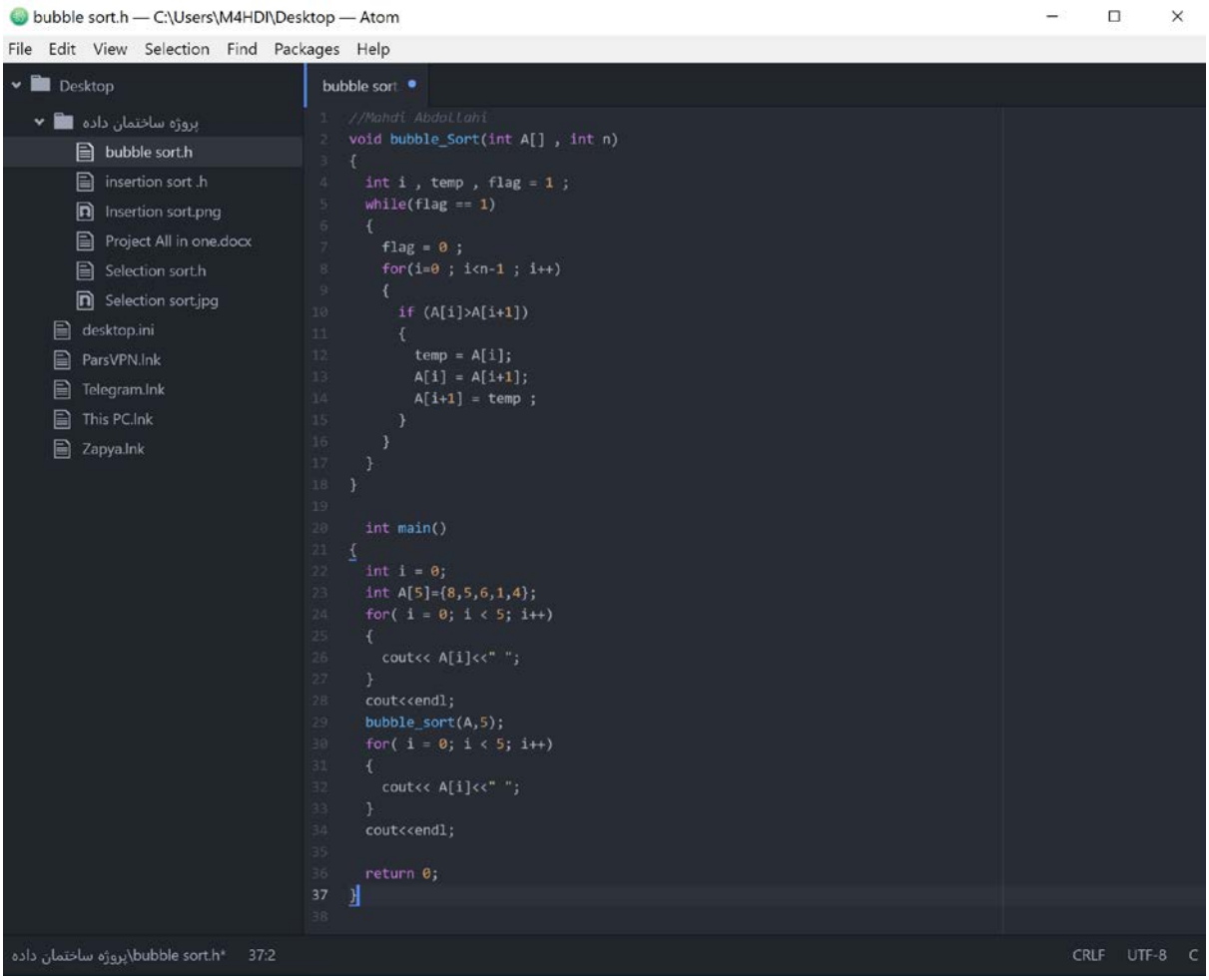
در این مرحله خانه آخر با خانه های قبل خود مقایسه شده و در جای مناسب خود قرار می گیرد.

۵- ۹ – ۱۳ – ۱۵ – ۲۰

- $Big\_O = O(n) = n^2$

مرتب سازی حبابی (**Bubble_Sort**) :

- الگوریتم:



- توضیح فارسی الگوریتم:

در این روش مرتب سازی هر خانه آرایه با خانه کناری خودش مقایسه می شود اگر عدد خانه n بزرگ تر از عدد خانه n+1 بود جای دو عدد خانه ها عوض می شود. این عمل تا زمانی ادامه پیدا می کند که همه آرایه ها به صورت یک لیست مرتب کنار یکدیگر قرار گیرند.

- مثال عددی:

۹ – ۴ – ۵ – ۷ – ۲

۴ – ۹ – ۵ – ۷ – ۲

۴ – ۵ – ۹ – ۷ – ۲

۴ – ۵ – ۷ – ۹ – ۲

۴ – ۵ – ۷ – ۲ – ۹

۴ – ۵ – ۲ – ۷ – ۹

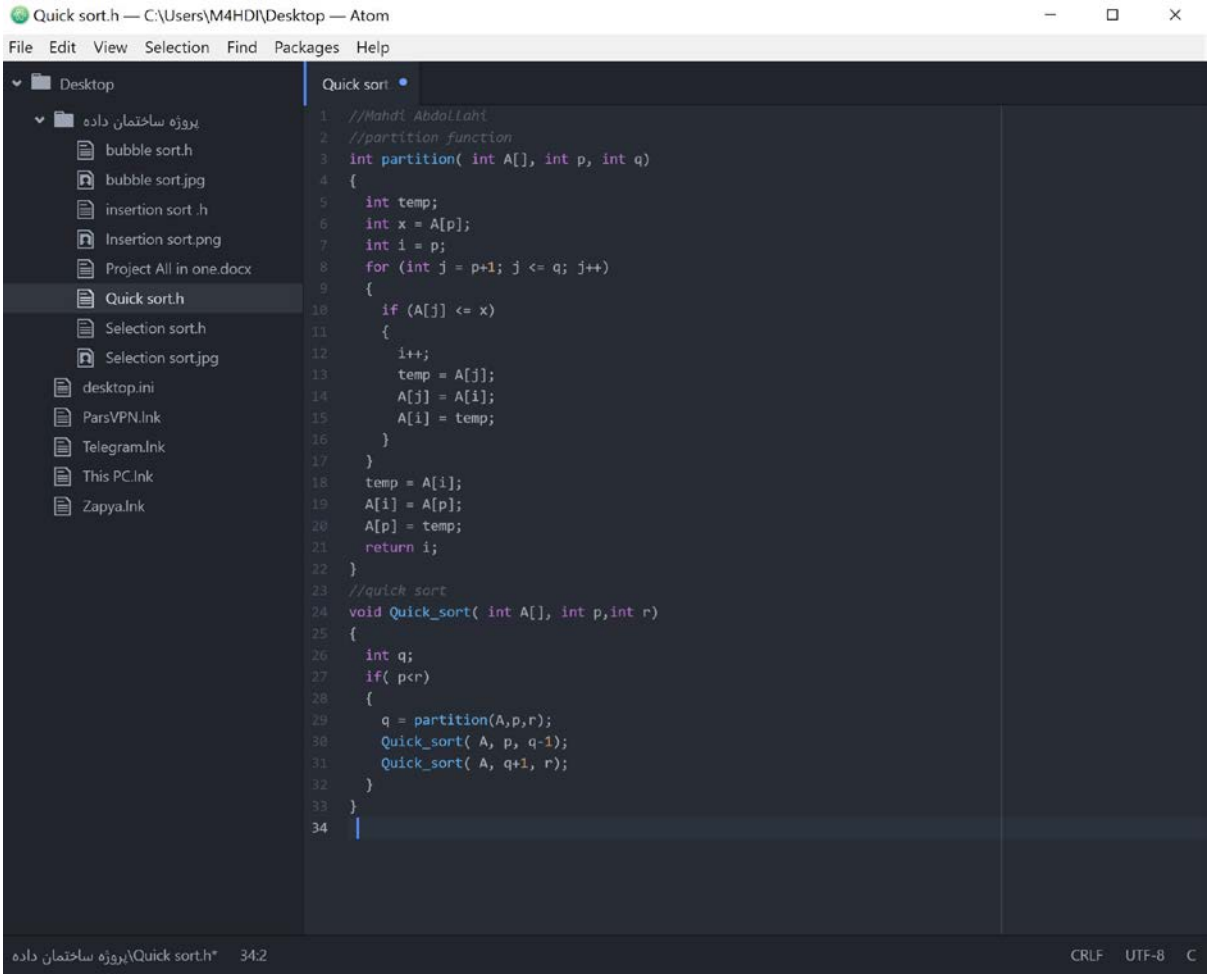
۴- ۲ – ۵ – ۷ – ۹

۲ – ۴ – ۵ – ۷ – ۹

- $Big\_O = O(n) = n^2$

مرتب سازی سریع (Quick_Sort) :

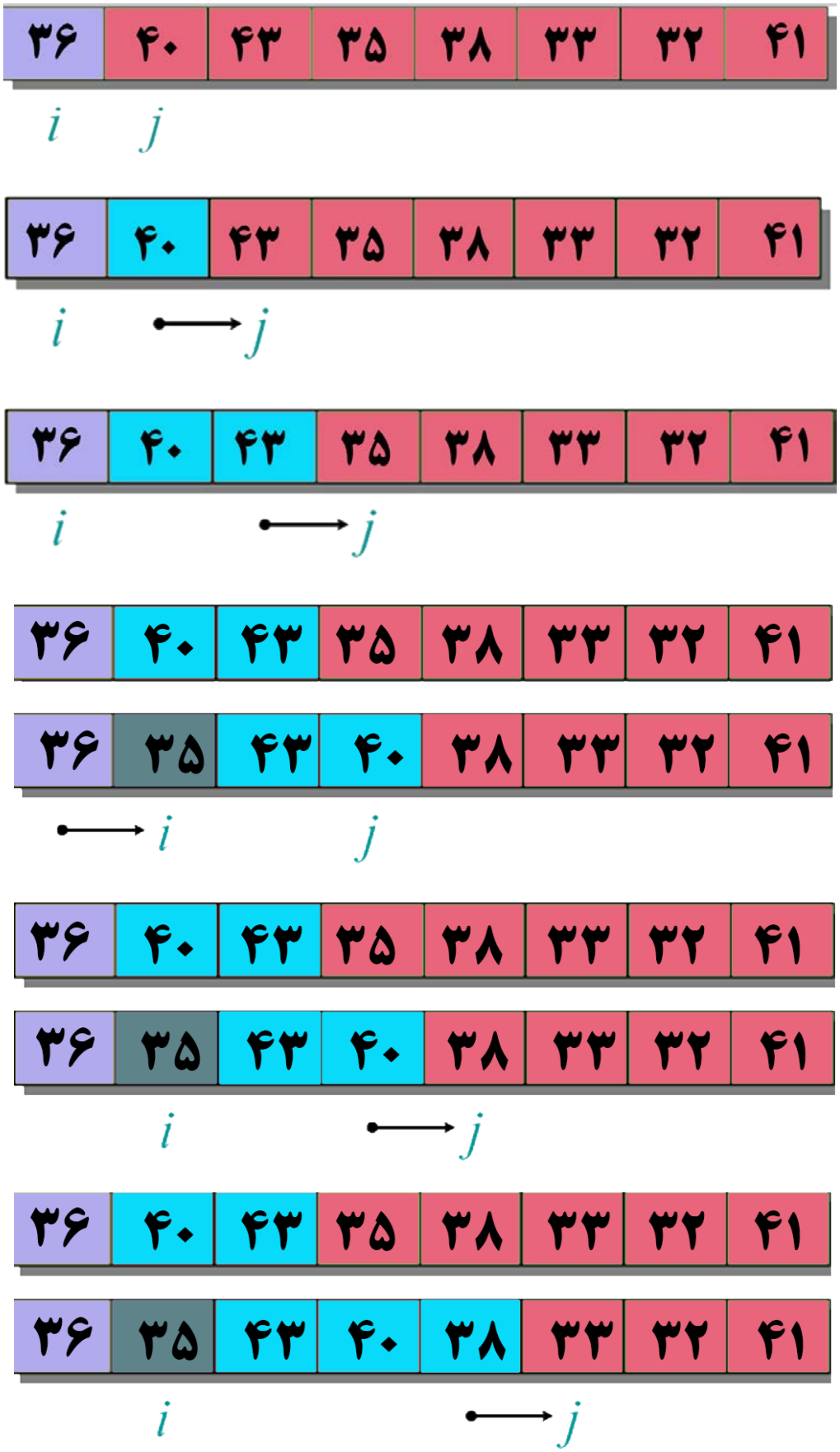
- **الگوریتم:**

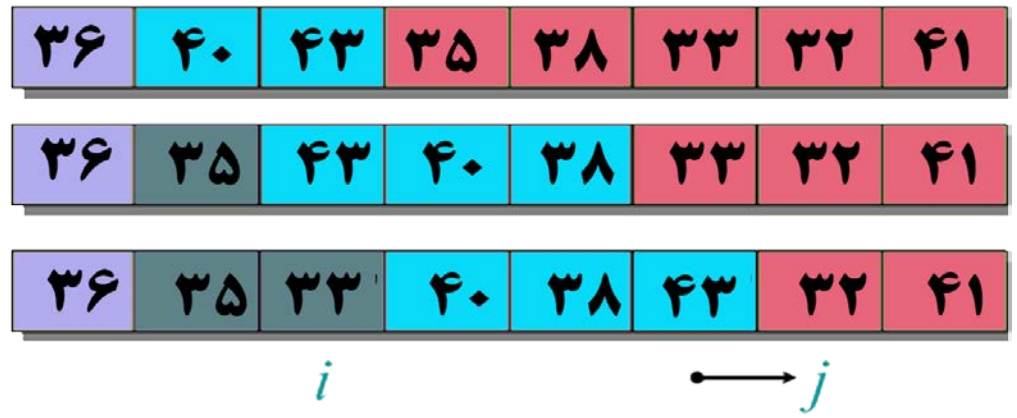


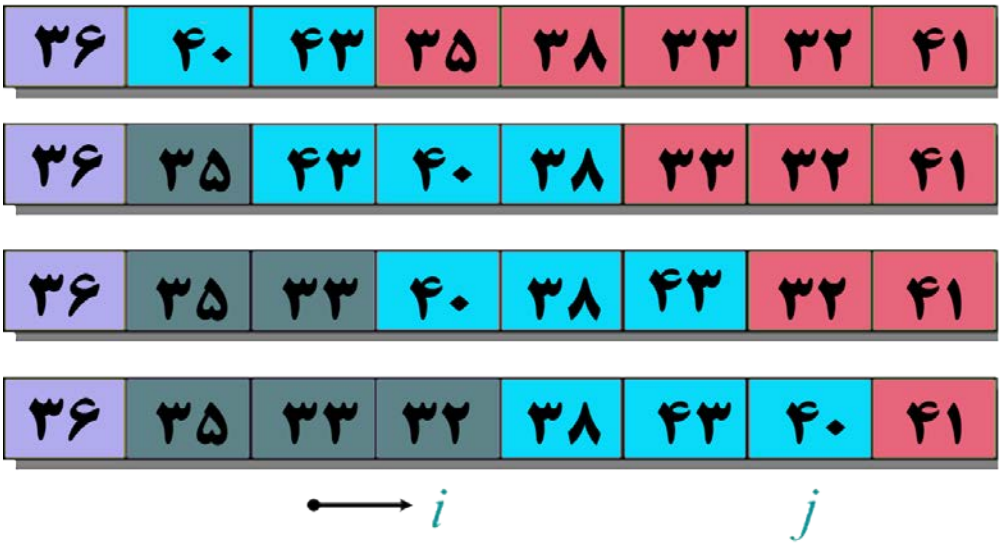
- توضیح فارسی الگوریتم:

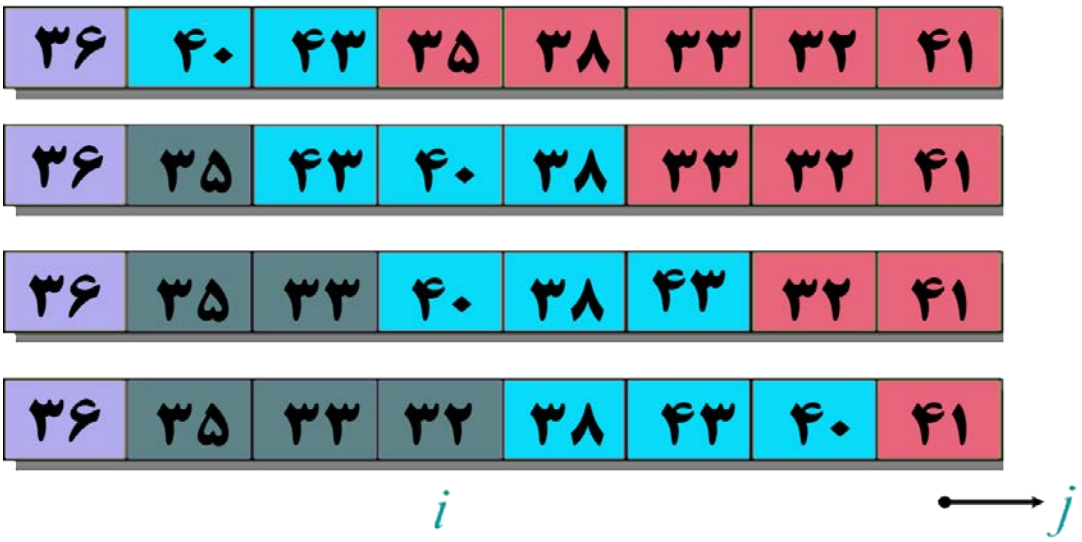
این تابع عناصرآرایه را به دوسته تقسیم می کند . آن هایی بزرگ تر از عنصر اول هستند و آن هایی که کوچکتر از عنصر اول هستند. و عنصر اول را بین این دو دسته قرار می دهد . و بعد مکانی را که عنصر ما بین این دو دسته قرار دارد را بر می گرداند.

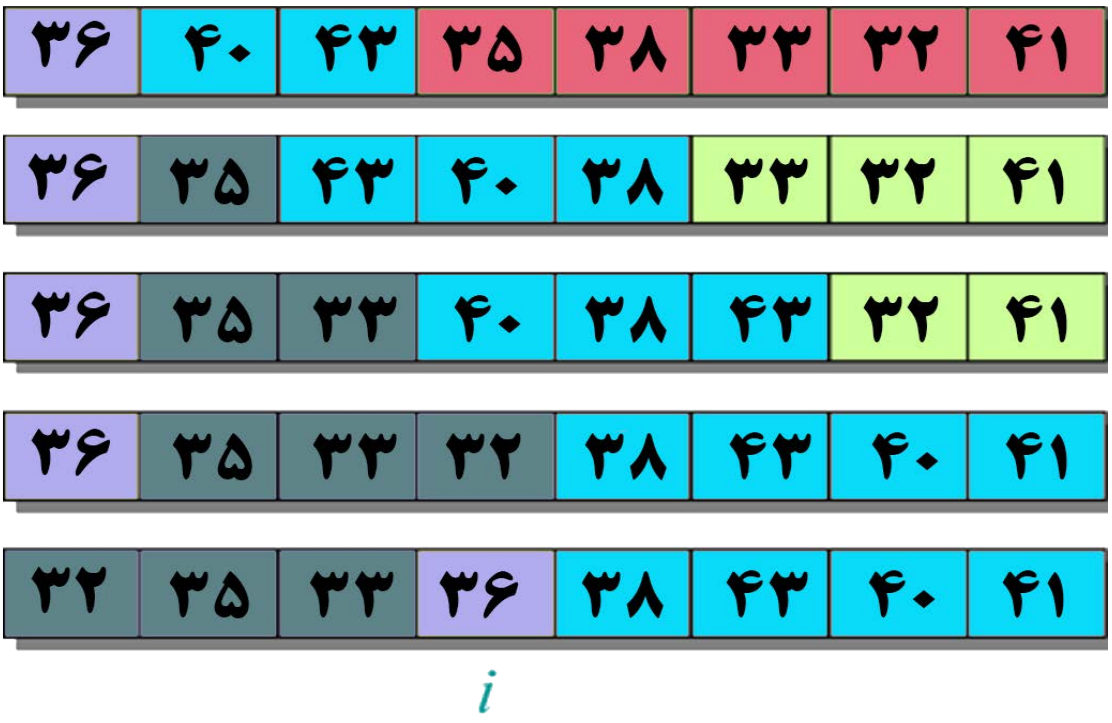
- مثال عددی:


۳۶ ۴۰ ۴۳ ۳۵ ۳۸ ۳۳ ۳۲ ۴۱
i j
۳۶ ۴۰ ۴۳ ۳۵ ۳۸ ۳۳ ۳۲ ۴۱
i j
۳۶ ۴۰ ۴۳ ۳۵ ۳۸ ۳۳ ۳۲ ۴۱
i j
۳۶ ۴۰ ۴۳ ۳۵ ۳۸ ۳۳ ۳۲ ۴۱
۳۶ ۳۵ ۴۳ ۴۰ ۳۸ ۳۳ ۳۲ ۴۱
i j
۳۶ ۴۰ ۴۳ ۳۵ ۳۸ ۳۳ ۳۲ ۴۱
۳۶ ۳۵ ۴۳ ۴۰ ۳۸ ۳۳ ۳۲ ۴۱
i j
۳۶ ۴۰ ۴۳ ۳۵ ۳۸ ۳۳ ۳۲ ۴۱
۳۶ ۳۵ ۴۳ ۴۰ ۳۸ ۳۳ ۳۲ ۴۱
i j


۳۶ ۴۰ ۴۳ ۳۵ ۳۸ ۳۳ ۳۲ ۴۱
۳۶ ۳۵ ۴۳ ۴۰ ۳۸ ۳۳ ۳۲ ۴۱
۳۶ ۳۵ ۳۳ ۴۰ ۳۸ ۴۳ ۳۲ ۴۱
i
j


۳۶ ۴۰ ۴۳ ۳۵ ۳۸ ۳۳ ۳۲ ۴۱
۳۶ ۳۵ ۴۳ ۴۰ ۳۸ ۳۳ ۳۲ ۴۱
۳۶ ۳۵ ۳۳ ۴۰ ۳۸ ۴۳ ۳۲ ۴۱
۳۶ ۳۵ ۳۳ ۳۲ ۳۸ ۴۳ ۴۰ ۴۱
i
j


۳۶ ۴۰ ۴۳ ۳۵ ۳۸ ۳۳ ۳۲ ۴۱
۳۶ ۳۵ ۴۳ ۴۰ ۳۸ ۳۳ ۳۲ ۴۱
۳۶ ۳۵ ۳۳ ۴۰ ۳۸ ۴۳ ۳۲ ۴۱
۳۶ ۳۵ ۳۳ ۳۲ ۳۸ ۴۳ ۴۰ ۴۱
i
j


٣٦ ٤٠ ٤٣ ٣٥ ٣٨ ٣٣ ٣٢ ٤١
٣٦ ٣٥ ٤٣ ٤٠ ٣٨ ٣٣ ٣٢ ٤١
٣٦ ٣٥ ٣٣ ٤٠ ٣٨ ٤٣ ٣٢ ٤١
٣٦ ٣٥ ٣٣ ٣٢ ٣٨ ٤٣ ٤٠ ٤١
٣٢ ٣٥ ٣٣ ٣٦ ٣٨ ٤٣ ٤٠ ٤١
i

- Big_O = O(n) =n^2