



**زیربرنامه:**

AscSortArray

|  |  |  |
| --- | --- | --- |
| **توسعه دهندگان** | مرتضی نامور |  |
| کامیار صفری |  |
| **تهیه کنندگان مستند** | مرتضی نامور، کامیار صفری | |
| **تاییدکنندگان** | مرتضی نامور | |
| **تاریخ تنظیم سند** | 28/5/1396 | |
| **شناسه سند** | **MC5F110F27** | |
| **زبان برنامه‌نویسی** | **Fortran 90** | |

1. وظایف

این زیربرنامه یک آرایه را دریافت کرده و آن را به‌صورت صعودی مرتب‌سازی می‌کند.

1. توضیحات و تئوری

این زیربرنامه یک آرایه را دریافت کرده و آن را به‌صورت صعودی از کوچک به بزرگ مرتب‌سازی می‌کند. روش مرتب‌سازی، مرتب‌سازی حبابی(Bubble Sort) می‌باشد. به این صورت که در هر مرحله، هر المان با المان بعدی مقایسه شده و درصورتی‌که از آن کوچک‌تر باشد، با یکدیگر تعویض می‌شوند. این الگوریتم تا زمانی که دیگر در یک گذر، در آرایه جابجایی بین المان‌ها صورت نگیرد، ادامه پیدا می‌کند. در این الگوریتم در هر گذر، بزرگ‌ترین المان در آخرین مکان مرتب شده قرار می‌گیرد و گذر بعدی به‌اندازه‌ی یک واحد کمتر صورت می‌گیرد به همین دلیل با توجه به اینکه در بهترین حالت ممکن است الگوریتم در یک گذر به آرایه‌ی مرتب شده برسد، الگوریتم در بهترین حالت از مرتبه‌ی O(n) بوده و با توجه به اینکه ممکن است تمامی گذرها لازم باشد به‌طور کامل اجرا شوند، مرتبه‌ی زمانی آن در بدترین حالت O(n2) می‌باشد.

1. بخش‌های زیربرنامه

در این قسمت، توضیح تمامی بخش‌های زیربرنامه، مطابق شماره‌گذاری انجام شده در متن برنامه کامپیوتری ارائه شده است.

1. پیمایش تمامی المان‌های آرایه‌ی ورودی و اجرای یک گذر به ازای هر المان

در این بخش، تمامی اعضای آرایه‌ی نامرتب ورودی پیمایش می‌شوند. با توجه به اینکه در پایان هرگذر بایستی بررسی کنیم که آیا جابجایی بین المان‌های آرایه صورت گرفته است یا خیر، یک متغیر را با مقدار FALSE مقداردهی اولیه کرده تا در صورت بروز جابجایی آن را TRUE کنیم. سپس در یک حلقه‌ی داخلی، هر گذر را به‌صورت یک حلقه به‌اندازه‌ی تعداد المان‌های باقیمانده‌ی مرتب‌سازی نشده اجرا می‌کنیم. در هر گذر، یکی از المان‌ها در مکان صحیح خود قرارگرفته و درنتیجه گذر بعد به‌اندازه‌ی یک واحد کمتر اجرا می‌شود.

1. مقایسه‌ی هر المان با المان بعدی و انجام جابجایی

در هر تکرار، هر المان آرایه‌ی نامرتب را با المان بعدی مقایسه کرده و درصورتی‌که از المان بعدی کوچک‌تر باشد، مکان آن‌ها را باهم تعویض می‌کنیم. همچنین برای اینکه در پایان گذر متوجه شویم جابجایی رخ‌داده است، متغیر مربوطه را با TRUE مقداردهی می‌کنیم. پس از پایان اجرای هر گذر، متغیری که انجام جابجایی را نشان می‌دهد را بررسی می‌کنیم. درصورتی‌که TRUE باشد به این معنی است که جابجایی رخ‌داده. درنتیجه آرایه ممکن است هنوز کامل مرتب‌نشده باشد. بنابراین باید مرتب‌سازی ادامه پیدا کند. در غیر این صورت اگر مقدار آن FALSE باشد، به این معنی است که هیچ جابجایی در گذر اجرا شده صورت نگرفته، درنتیجه آرایه کاملاً مرتب شده، پس نیازی به ادامه‌ی اجرای الگوریتم نیست، درنتیجه از حلقه خارج می‌شویم.