حالت پایه الگوریتم، آرایه ای به طول ۲ است که به درستی مرتب می شود. بنابر استقرا ریاضی فرض می کنیم که الگوریتم آرایه ای به طول $n-1$ را به درستی مرتب می کند. نشان می دهیم که الگوریتم آرایه به طول n را نیز مرتب خواهد کرد.
آرایه ای به طول n مانند $A = [a_0, a_1, \ldots, a_{n-1}]$ را در نظر می گیریم و با استفاده از الگوریتم دو سوم ابتدایی آرایه را مرتب می کنیم و داریم:

$$m = \lfloor \frac{n}{3} \rfloor \rightarrow A = [a_0, \ldots, a_{n-m}, \ldots, a_n]; a_0 < \ldots < a_m < \ldots < a_{n-m}$$

در مرحله ی بعدی دو سوم پایانی آرایه مرحله قبل را مرتب می کنیم و داریم :

$$m = \lfloor \frac{n}{3} \rfloor \rightarrow A = [a_0, \ldots, a_m, a'_{m+1}, \ldots, a'_n]; a'_{m+1} < \ldots < a'_{n-m} < \ldots < a'_n$$

در این مرحله اطمینان داریم که یک سوم پایانی این آرایه در جای درست خود قرار گرفته. برای اثبات این گزاره از برهان خلف استفاده می کنیم و فرض می کنیم که در یک سوم ابتدایی عددی وجود دارد که حداقل از یکی از اعداد یک سوم پایانی بزرگ تر است (از اعداد ۱ و ۳ برای نشان دادن یک سوم ابتدایی و پایانی آرایه به دست امده در مرحله دوم استفاده شده است):

$$\exists x \in 1 \;\; \exists y \in 3 : x > y \xrightarrow{x \leq a_m} \exists y \in 3 : a_m > y \xrightarrow{a_m < a_{m+1}} \exists y \in 3 : y < a_{m+1} < \ldots < a_{n-m} \rightarrow y \notin 3$$

نتیجه می گیریم y از یک سوم میانی آرایه پس از مرحله اول کوچکتر است و پس از مرحله دوم باید قبل از همه آن ها قرار بگیرد پس y نمی تواند بعد از مرحله دوم در یک سوم پایانی باشد که این تناقض است بنابر این حکم درست است و اعداد در یک سوم پایانی به درستی در جای خود قرار گرفته اند.
در مرحله سوم دوباره دو سوم ابتدایی آرایه را مرتب می کنیم و این باعث مرتب شدن تمام آرایه خواهد شد. بنابر استقرا درستی این الگوریتم برای آرایه های به طول n که n عددی صحیح و مثبت است اثبات شد.