ابتدا رابطه ترتیب را برای دو جایگشت P و P' تعریف می کنیم و داریم :

$$P_i = (A_{i0}, A_{i1}, \dots, A_{in-1})$$

$$P_j = (A_{j0}, A_{j1}, \dots, A_{jn-1})$$

فرض کنیم A_{im} اولین عنصری باشد که این دو جایگشت در آن باهم متفاوتند، گوییم $P_i > P_j$ اگر و تنها اگر $A_{im} > A_{jm}$ باشد. بنابرین اولین و آخرین جایگشت های این n تایی که کوچکترین و بزرگترین جایگشت ها نیز هستند را به صوت زیر داریم :

$$P_{first} = (p_0, p_1, \dots, p_{n-1}) \qquad \forall i > j : p_i > p_j$$
$$P_{last} = (p_0, p_1, \dots, p_{n-1}) \qquad \forall i > j : p_i < p_j$$

فرض کنیم در قسمتی از برنامه جایگشتی مانند $P = (a_0, a_1, \dots, a_{n-1})$ ایجاد شده است، حال رابطه بین P و P' را که جایگشت بعدی P است را پیدا می کنیم. فرض می کنیم اولین عنصری که P و P' که در آن باهم تفاوت دارند m امین عنصر آرایه باشد که آن را b می نامیم. بنابرین $b \neq a_{m-1}$ و از آنجایی که $P' > P$ نتیجه می گیریم $a_{m-1} < b$.

از آنجایی که P بزرگترین جایگشت بین جایگشت هایی است که m عنصر اول آنها با P برابر است می توانیم درباره $n - m$ عنصر باقیمانده نتیجه بگیریم که :

$$a_m > a_{m+1} > \dots > a_{n-1} \tag{1}$$

از طرفی $m - 1$ عنصر اول P' با P یکسان است و $b \neq a_{m-1}$ پس می توانیم نتیجه بگیریم که b یکی از $n - m$ عنصر باقیمانده خواهد بود که مجموعه آنها را S می نامیم :

$$S = \{a_m, a_{m+1}, a_{m+2}, ..., a_{n-1}\}$$

میدانیم که P' کوچکترین جایگشت از بین جایگشت های بزرگتر از P است پس b باید کوچکترین عضو S باشد به طوری که $b > a_{m-1}$ و $a_k = b$. چنین b زمانی وجود دارد که $a_m > a_{m-1}$.

حال با پیدا کردن b ، آن را با عنصر m ام جابجا می کنیم و $n - m$ و مجموعه جدید را S' می نامیم و داریم :

$$S' = (S \setminus \{a_k\}) \cup \{a_{m-1}\}$$

از آنجایی که P' از تمام جایگشت های بعد از خود کوچکتر است پس باید از اعضای آن از m به بعد به صورت نزولی مرتب شده باشند و از (1) داریم که اعضای S' به صورت صعودی مرتب شده اند پس کافی است ترتیب آن ها را برعکس کنیم تا به P' برسیم.

الگوریتم از P_{first} که اولین جایگشت است شروع می شود

حال بنابر استقرا فرض می کنیم که الگوریتم تا جایگشت P را محاسبه کرده است، نشان دادیم که الگوریتم با انجام مراحل بالا می تواند جایگشت بعدی P را تا هنگام برقرار بودن شرط محاسبه کند پس الگوریتم تمام جایگشت های این n تایی را از P_{first} تا P_{last} محاسبه می کند. بنابراین حکم ثابت شد و الگوریتم درست است.