فرض استقری تقویت شده: *می‌دانیم که چطور ضرایب توازن و ارتفاع‌های تمام گره‌های یک درخت با تعداد گره‌های کم‌تر از n را حساب کنیم.*

مجدداً حالت پایه بدیهی است. اکنون زمانی که ریشه را در نظر گرفتیم، می‌توانیم به سادگی ضریب توازن آن را با محاسبه‌ی تفاضل بین ارتفاع‌های فرزندان آن حساب کنیم. به علاوه می‌توان ارتفاع ریشه را نیز حساب کرد. این ارتفاع برابر بیشینه‌ی ارتفاع دو فرزند آن به علاوه‌ی ۱ است.

کلید حل این الگوریتم این است که مسأله‌ی اندکی گسترده‌تر را حل می‌کند، به‌جای این که فقط ضرایب توازن را حساب کنیم، ارتفاع‌ها را نیز حساب می‌کنیم. مسأله‌ی توسعه‌یافته، به نمونه‌ای ساده برای حل تبدیل می‌شود. زیرا محاسبه‌ی ارتفاع‌ها آسان است. در بسیاری از موارد حل مسأله‌ی قوی‌تر، آسان‌تر است! در استقرا تنها باید حل مسأله‌ی کوچک را به حل نمونه‌ی بزرگ‌تر بسط دهیم. اگر این حل وسیع‌تر باشد (زیرا مسأله بسط یافته است)، آنگاه احتمالاً گام استقرا آسان‌تر خواهد بود، زیرا چیزهای بیشتری برای کار کردن در اختیار داریم. این اشتباه رایجی است که گمان می‌رود در این مسأله دو پارامتر مجزا وجود داشته و هر کدام باید جداگانه حساب شوند. نمونه‌هایی از این اشتباه‌ها را بعداً در این کتاب نشان خواهیم داد.

۸.۵ یافتن بزرگ‌ترین زیردنباله‌ی متوالی

مسأله: دنباله‌ی $x_1, x_2, \ldots, x_n$ از اعداد حقیقی (نه الزاماً مثبت) داده شده است. دنباله‌ای چون $x_i, x_{i+1}, \ldots, x_j$ (شامل عنصرهای متوالی بیابید، به نحوی که حاصل‌جمع این اعداد در بین تمام زیردنباله‌های عنصرهای متوالی بیشتر باشد.

چنین زیردنباله‌ای را زیردنباله‌ی بیشینه می‌نامند. مثلاً در دنباله‌ی با اعداد به صورت (۲, −۳, ۱٫۵, −۱, ۳, −۲, −۳, ۳) زیردنباله‌ی بیشینه به صورت ۱٫۵, −۱, ۳ است که حاصل‌جمع آن برابر ۳٫۵ است. ممکن است چندین زیردنباله‌ی بیشینه در یک دنباله‌ی مفروض وجود داشته باشد. اگر تمام اعداد منفی باشند، زیردنباله‌ی بیشینه تهی است (با توجه به تعریف،

حاصل‌جمع زیردنباله‌ی تهی برابر صفر است). می‌خواهیم الگوریتمی بنویسیم که این مسأله را حل کرده و دنباله‌ی اولیه را تنها یک بار به ترتیب بخواند. فرض استقرای اولیه به صورت زیر است:

فرض استقرا: *می‌دانیم که چطور زیردنباله‌ی بیشینه را در دنباله‌ای با اندازه‌ی کمتر از n پیدا کنیم.*

اگر $n = ۱$ زیردنباله‌ی بیشینه تنها از همان یک عدد در صورتی که نامنفی باشد، تشکیل می‌شود و در غیر این صورت ۰ است. دنباله‌ای مانند $S = (x_۱, x_۲, \ldots, x_n)$ با اندازه‌ی $n > ۱$ را در نظر بگیرید. با استقرا می‌دانیم که چطور زیردنباله‌ی بیشینه را در $S' = (x_۱, x_۲, \ldots, x_{n-۱})$ بیابیم. اگر این زیردنباله‌ی بیشینه تهی باشد، آنگاه تمام اعداد S' منفی بوده و ما باید تنها x_n را در نظر بگیریم. فرض کنید که زیردنباله‌ی بیشینه یافته شده توسط استقرا در S' به صورت $S'_M = (x_i, x_{i+۱}, \ldots, x_j)$ است که i و j در رابطه‌ی $۱ \leq i \leq j \leq n - ۱$ صدق می‌کنند. اگر $j = n - ۱$ (یعنی زیردنباله‌ی بیشینه یک پسوند باشد) بسط جواب به S آسان است. اگر x_n مثبت باشد، آن را به S'_M اضافه می‌کنیم. در غیر این صورت آن را بدون تغییر باقی می‌گذاریم. اما اگر $j < n - ۱$ دو امکان وجود دارد. یا S'_M بیشینه باقی خواهد ماند یا زیررشته‌ی دیگری وجود خواهد داشت که در S'_M بیشینه نبوده، اما زمانی که عنصر x_n اضافه شد، در S بیشینه می‌شود. حل مسأله در گروی قوی‌تر کردن فرض استقراست. در ابتدا روشی را نشان می‌دهیم که با کمک آن می‌توان مسأله‌ی زیردنباله‌ی بیشینه را حل کرد و در بخش بعدی در مورد آن به صورتی کلی‌تر بحث خواهیم کرد. مشکلی که در مورد فرض استقرای اولیه با آن مواجه بودیم، این بود که x_n ممکن بود در زیردنباله‌ی S' بیشینه نبوده، ولی به زیردنباله‌ای اضافه شده و زیردنباله‌ی جدیدی ایجاد کند که بیشینه شود. بنابراین تنها اطلاع از زیردنباله‌ی بیشینه در S' کافی نیست، البته x_n تنها می‌تواند زیردنباله‌ای را گسترش دهد که در $n - ۱$ پایان یابد، یعنی پسوندی از S'. فرض کنید که فرض استقرا را قوی کرده‌ایم، به نحوه‌ی که یافتن پسوند بیشینه را شامل می‌شود. این پسوند را با $S'_E = (x_k, x_{k+۱}, \ldots, x_{n-۱})$ نشان می‌دهیم.

فرض استقرای تقویت شده: *در دنباله‌هایی به طول کم‌تر از n می‌دانیم که چطور یک زیردنباله‌ی بیشینه کلی و زیردنباله‌ی بیشینه پسوند را بیابیم.*

```
Algorithm Maximum_Consecutive_Subsequence (X,n);
Input: X (an array of size n).
Output: Global_Max (the sum of the maximum subsequence).

begin
   Global_Max := 0;
   Suffix_Max := 0;
   for i := 1 to n do
       if x[i] + Suffix_Max > Global_Max then
          Suffix_Max := Suffix_Max + x[i];
          Global_Max := Suffix_Max
       else if x[i] + Suffix_Max > 0 then
               Suffix_Max := x[i] + Suffix_Max
            else Suffix_Max := 0
end
```

شکل ۹.۵: الگوریتم $Maximum\_Consecutive\_Subsequence$

اگر این دو زیردنباله را داشته باشیم، الگوریتم واضح می‌شود. کافیست x_n را به پسوند بیشینه اضافه کنیم. اگر حاصل‌جمع آن بیشتر از حاصل‌جمع زیردنباله‌ی بیشینه کلی شد، آنگاه زیردنباله‌ی بیشینه جدیدی خواهیم داشت (و هم چنین یک پسوند جدید). در غیر این صورت با همان زیردنباله‌ی بیشینه قبلی ادامه می‌دهیم. کار ما هنوز پایان نیافته است. باید هم‌چنین پسوند بیشینه جدیدی را بیابیم. همیشه نمی‌توان به سادگی x_n را به پسوند بیشینه قبلی اضافه کرد. ممکن است پسوند بیشینه که در x_n پایان می‌یابد، منفی شود. در این حالت بهتر است که این پسوند بیشینه را به صورت مجموعه‌ی تهی در نظر بگیریم (بعداً x_{n+1} به تنهایی در نظر گرفته خواهد شد). الگوریتم یافتن زیردنباله‌ی بیشینه در شکل ۹.۵ ارائه شده است.

۹.۵ قوی کردن فرض استقرا

قوی‌تر کردن فرض استقرا یکی از مهم‌ترین روش‌هایی است که برای اثبات قضایای ریاضی به کمک استقرا استفاده می‌شود. زمان انجام یک اثبات استقرایی غالباً با چنین وضعی مواجه می‌شویم. P نشان‌دهنده‌ی قضیه است. فرض استقرا توسط $P(< n)$ مشخص شده و اثبات باید این نتیجه را بدهد که $P(< n) \Rightarrow P(n)$. در بسیاری از موارد می‌توانیم فرض دیگری را با نام Q اضافه کنیم که با کمک آن اثبات آسان‌تر می‌شود. یعنی اثبات $[P\ and\ Q](< n) \Rightarrow P(n)$ نسبت به $P(< n) \Rightarrow P(n)$ آسان‌تر است. این فرض درست به نظر می‌رسد، اما مشخص نیست که چطور می‌توانیم آن را اثبات کنیم. ترفندی که می‌توانیم از آن استفاده کنیم این است که Q را در فرض استقرا بگنجانیم. اکنون باید ثابت کنیم که $[P\ and\ Q](< n) \Rightarrow [P\ and\ Q](n)$. P و Q قضایای قوی‌تری نسبت به P ی تنها هستند. اما اغلب اثبات قضایای قوی‌تر آسان‌تر است! این روند می‌تواند تکرار شود و با افزودن فرض‌هایی درستی اثبات مورد نظر حاصل شود. مسأله‌ی بزرگ‌ترین زیردنباله، مثال خوبی در مورد کاربرد این اصل برای بهبود الگوریتم‌هاست. تشابه زیبایی بین این اصل و عبارت زیر وجود دارد. «افزودن سود یک میلیون تومانی به فروش ۱۰۰ میلیون تومانی آسان‌تر از افزودن هزار تومان سود برای فروش ۱۰ تومانی است.»

اشتباه رایجی که هنگام استفاده از این شیوه می‌شود، فراموش کردن اثبات فرضِ اضافه شده است. به بیان دیگر آن‌ها ثابت می‌کنند که $[P\ and\ Q](< n) \Rightarrow P(n)$، حتی مفروض بودن Q را متذکر هم نمی‌شوند. این اشتباه متناظر با فراموش کردن محاسبه‌ی زیردنباله‌ی بیشینه پسوند در مسأله‌ی زیردنباله‌ی بیشینه است. در مثال ضریب توازن، مطلب فوق معادل با عدم محاسبه‌ی ارتفاع‌ها به طور مستقل است. متاسفانه این اشتباه بسیار رایج است. بیش از این روی این عبارت تأکید نمی‌کنیم که:

باید از فرض استقرا به صورت کاملاً دقیق پیروی کرد.

مثال‌های پیچیده‌تری را در مورد قوی‌تر کردن فرض استقرا از جمله در بخش‌های ۳.۱۱.۶، ۱.۱۳.۶، ۵.۷، ۳.۸ و ۱.۳.۱۲ ارائه خواهیم داد.