Data Structures and algorithms
Instructor : **Dr.Amin Eskandari**
Head-Ta's : Hamid Namjoo manesh , Amir Hossein Hemati , Ali Mojahed
Ta's : Amir Mohammad Asadjoo , Padina razmjooi , Armin Janfaza , Alireza Heydari , Mahdi Torabi ,
Maryam Ghorbani , Fatemeh Keshavarz , MohammadReza Fasli , Abolfazl Tadrisi

Week 03 Answers

پاسخنامه تکالیف هفته ۰۳

پاسخ سوال اول (۱۰ امتیاز) :

ایده و روش حل مسئله:

در این مسئله باید یک صف (Queue) را شبیه‌سازی کنیم .صف یک ساختار داده از نوع FIFO است؛ یعنی اولین وارد شده، اولین خارج می‌شود .بنابراین هر مشتری/عنصر با دستور ENQUEUE به انتهای صف اضافه می‌شود و با دستور DEQUEUE از ابتدای صف حذف می‌گردد.

نحوه اجرای دستورات:

1) ENQUEUE x : مقدار x به انتهای صف اضافه می‌شود.
2) DEQUEUE : عنصر ابتدای صف حذف می‌شود و همان عنصر چاپ می‌گردد .اگر صف خالی باشد باید پیام زیر چاپ شود : Empty queue
3) DISPLAY : وضعیت فعلی صف از ابتدا تا انتها چاپ می‌شود .اگر صف خالی باشد باید [] چاپ شود.
4) EXIT : اجرای برنامه خاتمه می‌یابد و پس از آن هیچ خروجی دیگری تولید نمی‌شود.

بررسی حالت‌های مرزی:(Edge Cases)

•اگر چند بار پشت سر هم DEQUEUE اجرا شود و صف خالی شود، از آن به بعد برای هر DEQUEUE باید فقط پیام Empty queue چاپ شود.
•اگر DISPLAY در هر لحظه‌ای فراخوانی شود، باید دقیقاً وضعیت همان لحظه صف را نشان دهد.
•بعد از EXIT نباید هیچ خروجی‌ای چاپ شود.

مدیریت ورودی نامعتبر:

اگر دستور ناشناخته باشد یا تعداد پارامترهای دستور درست نباشد، می‌توان پیام "Invalid input" چاپ کرد و برنامه به خواندن دستورات بعدی ادامه دهد .این بخش وابسته به صورت دقیق صورت‌سؤال/قوانین تعریف‌شده است.

تحلیل پیچیدگی:

•ENQUEUE : در حالت معمول(1)O
• DEQUEUE : اگر با اشاره‌گر ابتدای صف پیاده‌سازی شود، به طور میانگین(1)O
• DISPLAY : به دلیل چاپ تمام عناصر صف،(n)O
فضای مصرفی نیز متناسب با تعداد عناصر صف یعنی (n)O است.

پاسخ سوال دوم (۳۰ امتیاز) :

در این مسئله باید صف بیماران را طوری مدیریت کنیم که بیمار عادی به انتهای صف اضافه شود (رفتار صف معمولی)، اما بیمار اورژانسی به ابتدای صف اضافه شود .برای این رفتار، بهترین ساختار داده صف دوطرفه (Deque)است چون هم اضافه‌کردن در ابتدا و هم در انتها را به‌صورت مستقیم پشتیبانی می‌کند.

تعریف عملیات‌ها و اثر آن‌ها:

1) ARRIVE_NORMAL x : بیمار x به انتهای صف اضافه می‌شود.
2) ARRIVE_URGENT x : بیمار x به ابتدای صف اضافه می‌شود.
3) SERVE : اولین بیمار صف حذف می‌شود و نام او چاپ می‌گردد .اگر صف خالی باشد پیام زیر چاپ می‌شود:
No patient
4) DISPLAY : وضعیت فعلی صف از ابتدا تا انتها نمایش داده می‌شود .اگر صف خالی باشد [] چاپ می‌شود.
5) EXIT : برنامه پایان می‌یابد و پس از آن خروجی دیگری تولید نمی‌شود.

•شما باید تفاوت Queue و Deque را بدانید و متوجه باشید که اورژانسی‌ها از ابتدا اضافه می‌شوند.
•ترتیب سایر بیماران باید حفظ شود و فقط محل اضافه‌شدن ابتدا/انتها متفاوت است.
•مدیریت حالت خالی بودن صف در SERVE و DISPLAY ضروری است.

حالت‌های مرزی:(Edge Cases)

•اگر چند بار پشت سر هم SERVE اجرا شود و صف خالی شود، از آن به بعد باید فقط No patient چاپ شود.
•اگر فقط اورژانسی‌ها وارد شوند، ترتیبشان معکوس ورود خواهد شد چون هر بار به ابتدای صف اضافه می‌شوند.
•اگر DISPLAY در هر لحظه فراخوانی شود، باید وضعیت همان لحظه صف را نشان دهد.

مدیریت ورودی نامعتبر(در صورت نیاز)

اگر دستور ناشناخته باشد یا تعداد پارامترها درست نباشد، می‌توان پیام "Invalid input" چاپ کرد و برنامه به خواندن دستورات بعدی ادامه دهد .این بخش بسته به قوانین دقیق صورت سؤال قابل تنظیم است.

تحلیل پیچیدگی:

•افزودن به ابتدا یا انتها در Deque معمولاً O(1) است.
•حذف از ابتدای Deque نیز O(1) است.
DISPLAY •به دلیل چاپ همه عناصر، O(n) است.
فضای مصرفی متناسب با تعداد بیماران حاضر در صف یعنی O(n) است.

پاسخ سوال سوم (۵۰ امتیاز) :

۱) بررسی اعتبار ساختار درخت:

ریشه در اندیس 0 مقدار A دارد، بنابراین شرط اول برقرار است .با بررسی تمام اندیس‌ها مشاهده می‌شود که هر گره‌ای که مقدار None دارد، در زیرشاخه خود دارای گره غیر None نیست .بنابراین ساختار درخت معتبر است.

۲) استخراج پیمایش‌ها:

Preorder : ریشه، چپ، راست
[A, B, D, C, E, F, G]

Inorder : چپ، ریشه، راست
[D, B, A, E, C, G, F]

Postorder : چپ، راست، ریشه:
[D, B, E, G, F, C, A]

۳) محاسبه ارتفاع درخت:

بیشترین عمق درخت برابر با مسیر A → C → F → G است که شامل 4 سطح می‌باشد .بنابراین ارتفاع درخت برابر با 4 است.

پاسخ سوال چهارم (۷۰ امتیاز) :

در این مسئله یک شمارنده دودویی به‌صورت آرایه‌ای از بیت‌ها نگهداری می‌شود .برای افزایش شمارنده، از بیت کم‌ارزش شروع کرده و عملیات کری را مشابه جمع دودویی انجام می‌دهیم.

نحوه محاسبه هزینه هر مرحله:

در هر عملیاتINCREMENT ، هر بار که یک بیت از ۰ به ۱ یا از ۱ به ۰ تغییر می‌کند، یک واحد هزینه در نظر گرفته می‌شود .بنابراین هزینه هر مرحله برابر با تعداد بیت‌هایی است که مقدار آن‌ها تغییر کرده است.

تحلیل سرشکن:(Amortized Analysis)

اگرچه در برخی مراحل ممکن است تعداد زیادی بیت تغییر کنند مثلاً هنگام تبدیل 01111 به10000 ، اما در مجموع و به‌طور میانگین، هزینه هر عملیات INCREMENT ثابت و از مرتبه O(1) است .این نتیجه نشان‌دهنده‌ی کارایی مناسب شمارنده دودویی در تحلیل سرشکن است.

نتیجه‌گیری:

شما باید بتوانید هم منطق دودویی افزایش شمارنده را پیاده‌سازی کنید و هم مفهوم هزینه سرشکن را درک کرده و محاسبه نمایید.

پاسخ سوال پنجم (۱۰۰ امتیاز) :

در این مسئله باید از دو ساختار داده به‌صورت هم‌زمان استفاده شود :صف (Queue) برای نگهداری ترتیب ورود پهپادها و پشته (Stack) برای ذخیره موقت پهپاد در شرایط اضطراری .صف رفتار FIFO دارد و پشته رفتار.LIFO

توضیح دستور EMERGENCY نکته کلیدی سؤال

در این دستور، اگر حداقل دو پهپاد در صف وجود داشته باشد، اولین پهپاد به طور موقت از صف خارج شده و داخل پشته ذخیره می‌شود .سپس پهپاد دوم صف عبور می‌کند و حذف می‌شود .در نهایت، پهپاد ذخیره‌شده از پشته خارج شده و باید دوباره به ابتدای صف بازگردد .این کار بدون استفاده از پشته به‌درستی قابل انجام نیست و هدف اصلی سؤال، درک تفاوت FIFO و LIFO است.

مدیریت دستورات دیگر:

- ARRIVE : افزودن پهپاد به انتهای صف.
- PASS : حذف پهپاد ابتدای صف؛ اگر صف خالی باشد پیام Empty queue چاپ می‌شود.
- DISPLAY : نمایش وضعیت فعلی صف از ابتدا تا انتها.
- EXIT : پایان برنامه بدون تولید خروجی اضافی.

حالت‌های مرزی:(Edge Cases)

•اگر EMERGENCY زمانی اجرا شود که کمتر از دو پهپاد در صف باشد، هیچ تغییری ایجاد نمی‌شود.
•اگر چند بار PASS روی صف خالی اجرا شود، هر بار باید پیام Empty queue چاپ شود.
•ترتیب پهپادهای باقی‌مانده در صف نباید در هیچ حالتی به‌هم بخورد.

نکات کلیدی سوال:

•نیاز به استفاده هم‌زمان از دو ساختار داده متفاوت دارد.
•درک عمیق از تفاوت FIFO و LIFO ضروری است.
•پیاده‌سازی دستور EMERGENCY بدون درک مفهومی درست به خطا منجر می‌شود.