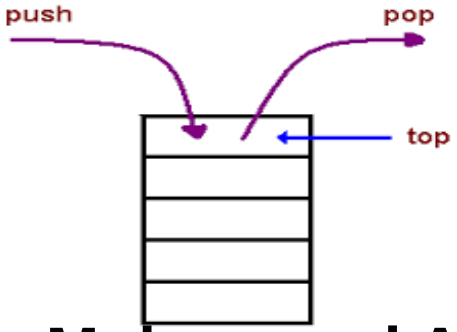
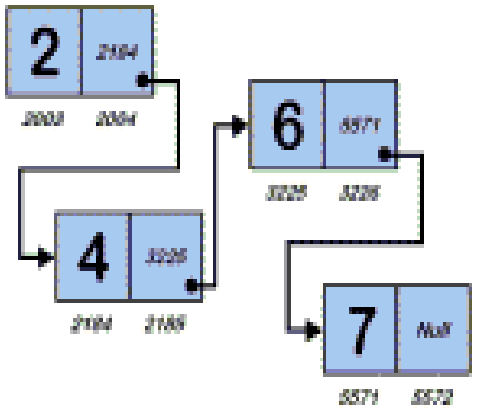
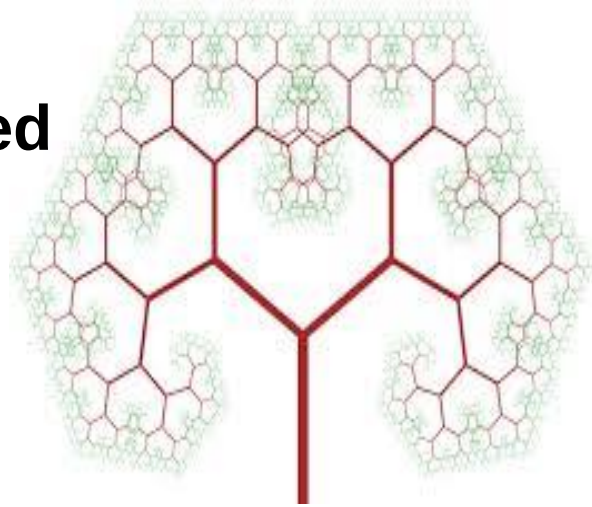
جامعة حماة
الكلية التطبيقية
قسم تقنيات الحاسوب
الفصل الثاني



- المقرر: الخوارزميات وبنى المعطيات
- المحاضرة :التاسعة

Dr. Mohammed AL-Mohammed





Chapter 9

بنى المعطيات

Data structure

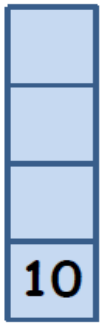
الأرتال والمكدسات

الرتل(صف الانتظار) Queue

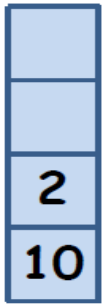
الأرتال والمكدسات

From Postfix to Answer

Ex: 10 2 8 * + 3 -



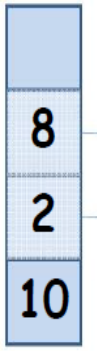
- First, push(10) into the stack



- Then, push(2) into the stack

- Push(8) into the stack

- Now we see an operator *, that means we can get an new number by calculation



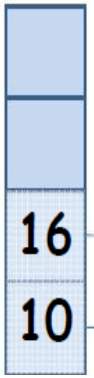
- Pop the first two numbers

2 * 8 = 16

- Push the new number back

2 * 8 = 16

Ex: 10 2 8 * + 3 -



- Then we see the next operator + and perform the calculation

10 + 16 = 26

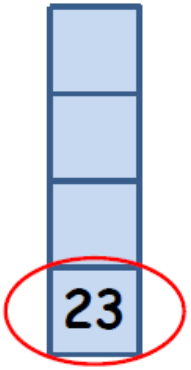
- Push the new number back

10 + 16 = 26

- We see the next number 3
- Push (3) into the stack

- The last operation

26 - 3 = 23



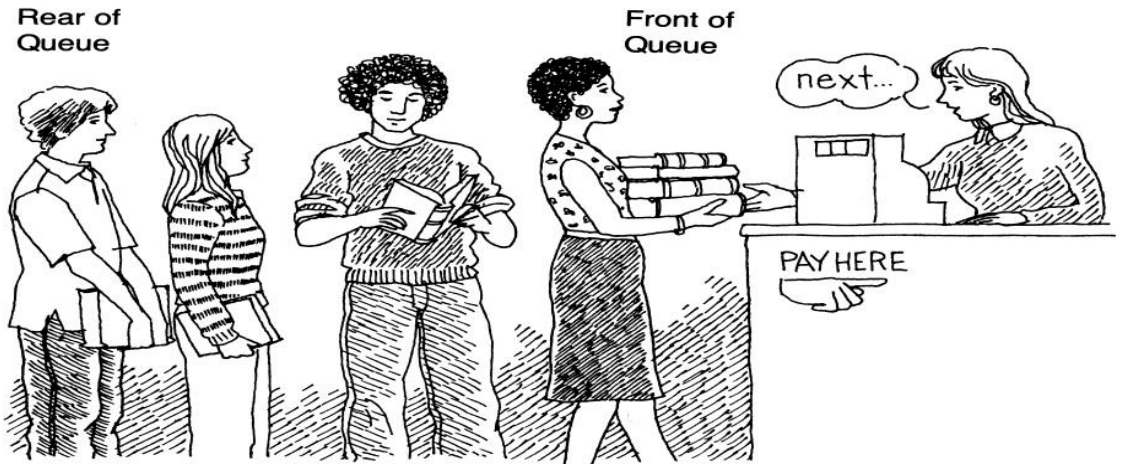
26 - 3 = 23

answer!

What is a queue?

- Stores a set of elements in a particular order
- Queue principle: FIRST IN FIRST OUT = (FIFO)
- It means: the first element inserted is the first one to be removed

➢ Example



- The first one in line is the first one to be served

مميزات استخدام المصفوفة بالطابور:

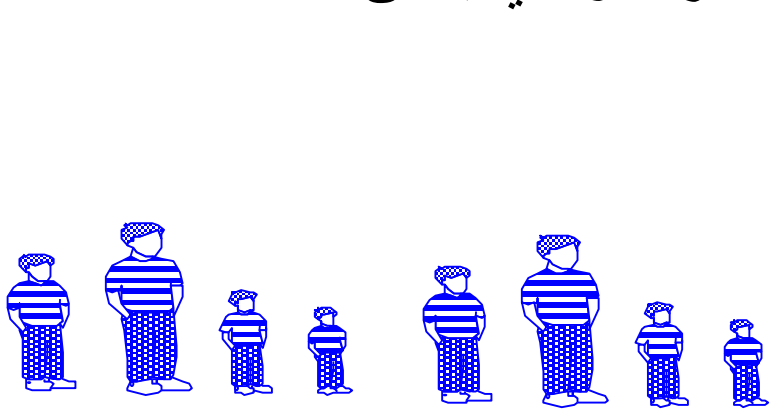
- تنفيذها بسيط.
- يجب ان يحدد حجم الطابور عند الاعلان.
- الفضاء ممكن ان لايستخدم بشكل جيد ويكون فقدان بمواقع الذاكرة عند استخدام عدد قليل من العناصر مع الاعلان عن حجم كبير للطابور.
- لايمكن اضافة عناصر اضافية اكثر من عدد العناصر التي حدد بها حجم الطابور (حجم المصفوفة).

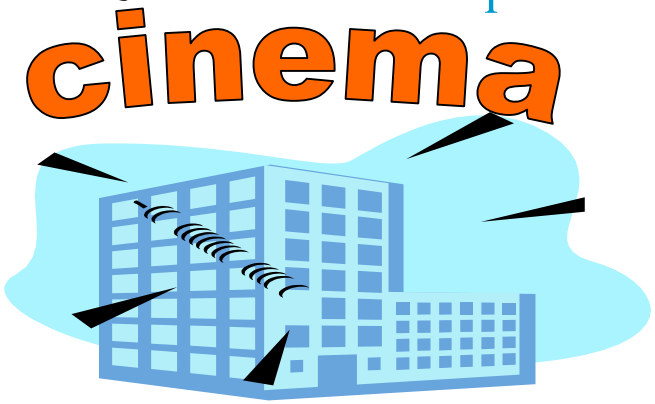
مميزات تنفيذ الطابور باستخدام القوائم الموصولة:

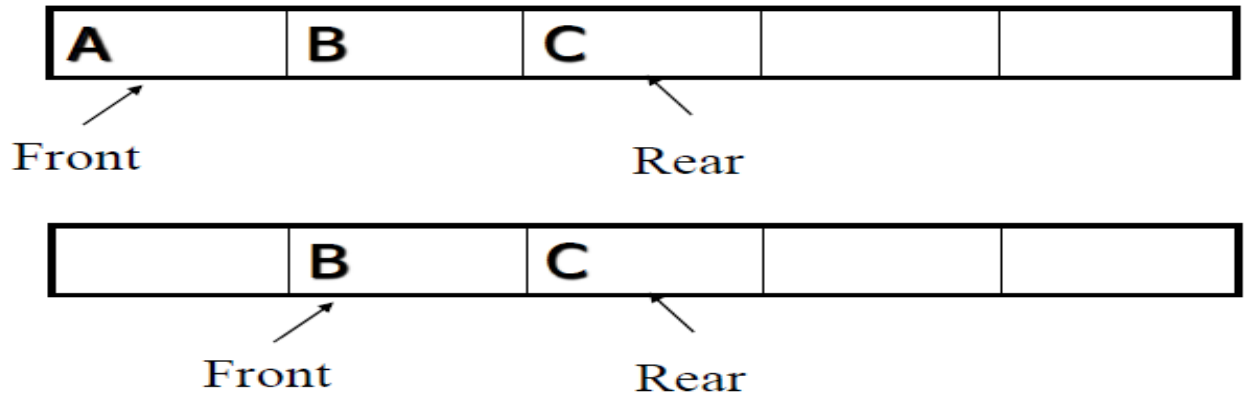
- تخصيص ذاكرة بشكل الي لكل عنصر جديد.
- ربط عناصر الطابور مع بعضها بواسطة المؤشرات.
- استخدام مؤشرين واحد يؤشر على العنصر الاول بالطابور والثاني يؤشر على العنصر الاخير بالطابور.
- تعتبر طريقة جيدة لترشيد استهلاك الذاكرة.

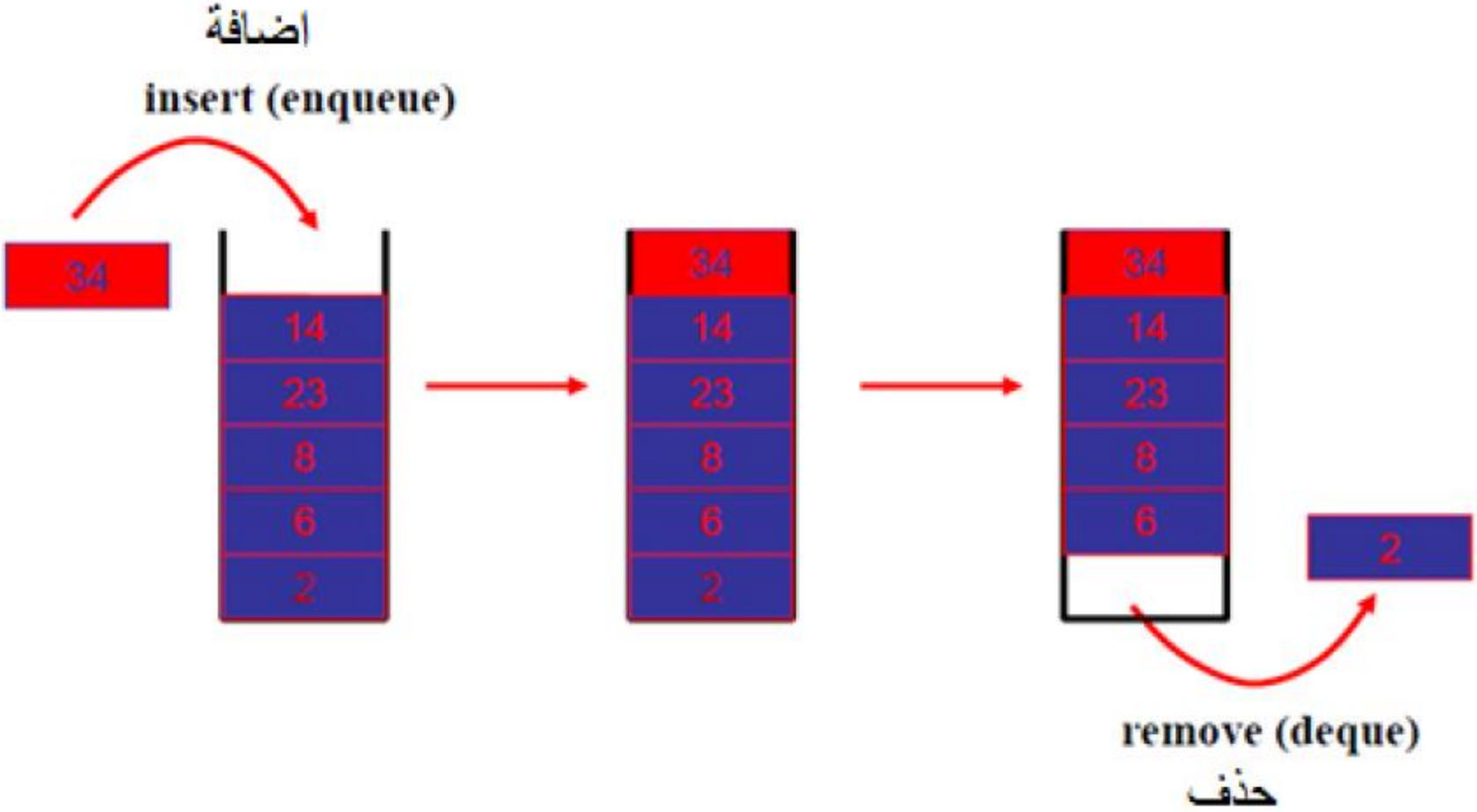
النمط المجرد "الرتل Queue"

- العمليتان الأساسيتان هما Enqueue وDequeue:
  - Enqueue تدرج عنصراً في نهاية الرتل والذي يسمى rear.
  - Dequeue تحذف عنصراً من بداية الرتل والذي يسمى front.



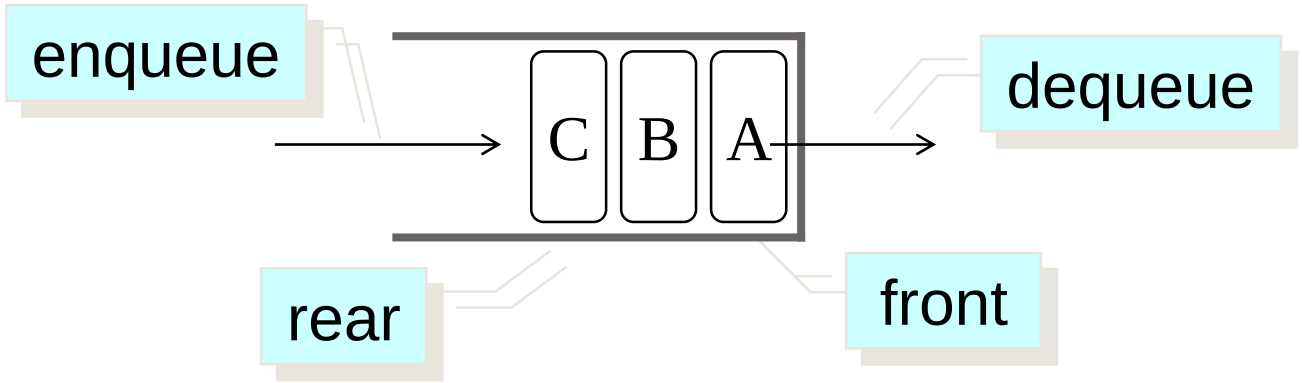






عمليات الاضافة والحذف في الطابور الخطي

الرتل



- الرتل مفتوح من جهتين.
- يمكن إضافة عناصر (enqueue) فقط عند آخره rear، وحذف العناصر (dequeue) عند أول الرتل front.
- ليس بالإمكان إضافة/استخراج العناصر من داخل الرتل.
- يجري إخراج العناصر من الرتل وفق ترتيب الإدخال نفسه (بخلاف المكدس)

Queues vs.stacks

- queue is an ordered collection of items from which items may be deleted at one end (called the front of the queue) and into which items may be inserted at the other end (called the rear of the queue)

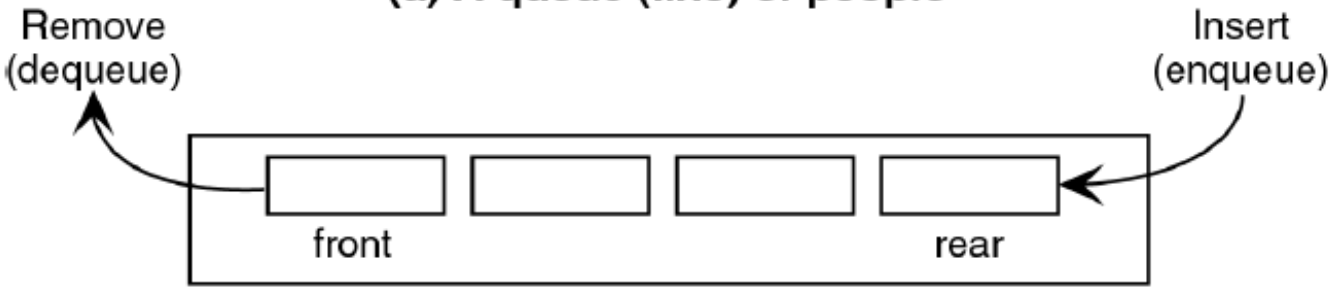
- In a stack, all insertions and deletions occur at one end, the **top**, of the list(LIFO (last in, first out)).
- In the queue all deletions occur at the head of the list.

However, all insertions to the queue occur at the tail of the list

Introduction to Queues

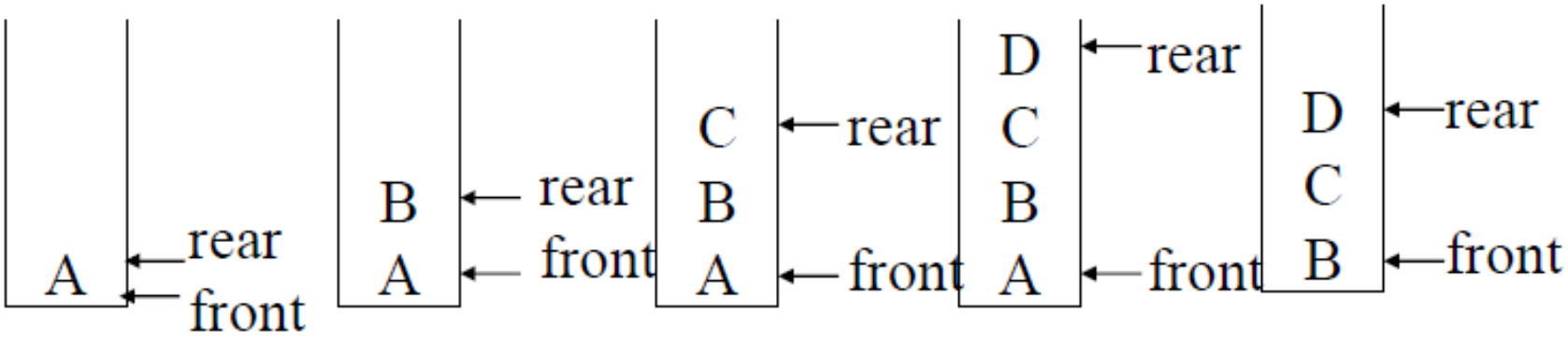
Basically, data enters the queue at one end and exits at the other end.

(a) A queue (line) of people



(b) A computer queue

First In First Out



طرائق نوع المعطيات المجرد Queue

- يدعم الرتل طريقتين أساسيتين:
- enqueue(o): يدرج الغرض *o* في آخر *rear* الرتل

  الدخل: غرض؛ ***الخرج***: لا شيء
- dequeue(): تحذف الغرض من أول الرتل وتعيده؛ ويحدث خطأ إذا كان الرتل فارغاً.

  الدخل: لاشيء؛ ***الخرج***: غرض

طرائق نوع المعطيات المجرد - تابع

- ويمكن تعريف هذه الطرائق الداعمة أيضاً:
  - size(): تعيد عدد الأغراض في الرتل.

    الدخل: لاشيء؛ ***الخرج***: عدد صحيح
  - isEmpty(): تعيد بولياناً يدل إذا كان الرتل فارغاً أم لا.

    الدخل: لاشيء؛ ***الخرج***: بولياني
  - front(): يعيد، ولكن لا يحذف، الغرض الموجود في أول الرتل؛ ويحدث خطأ إذا كان الرتل فارغاً.

    الدخل: لاشيء؛ ***الخرج***: غرض

Applications

- Ticketing counter
- Bus stop line
- Bank Customers
- Job scheduling (e.g. Round-Robin algorithm for CPU allocation)

Applications: Job Scheduling

front	rear	Q[0]	Q[1]	Q[2]	Q[3]	Comments
-1	-1					queue is empty
-1	0	J1				Job 1 is added
-1	1	J1	J2			Job 2 is added
-1	2	J1	J2	J3		Job 3 is added
0	2		J2	J3		Job 1 is deleted
1	2			J3		Job 2 is deleted

تطبيقات الأرتال

- تُستخدَم الأرتال عموماً لتنظيم استخدام مورد وحيد مشترك بين مجموعة من المستخدمين.
- الأرتال شائعة كثيراً في نظم تشغيل الحواسيب وفي الشبكات:
  - الرتل الخاص بالطابعة printer_queue
  - الرتل الخاص بالإجرائيات processes_queue
  - قناة الدخل/الخرج (I/O buffer)
  - الكتابة/القراءة على القرص الصلب
  - ...

بعض تطبيقات الطابور:

1. عندما يكون لدينا مصادر مشتركة مثل طابعة تستلم اوامر طباعة من اكثر من حاسوب او تستلم من حاسوب واحد اكثر من امر بالطباعة, في هذه الحالة فان امر طباعة واحد سيتم تنفيذة بينما الاوامر الاخرى تنتظر دورها للتنفيذ, ويتم تنفيذها حسب اسبقية وصولها (او وفقا لخوارزمية محددة بالتنفيذ).
2. ا ذا ماعملنا على الحاسوب وعملنا ضغط سريع بالماوس على اكثر من ملف بفارق وقت قليل جدا, وكان الملف الاول يحتاج الى بعض الوقت لانجاز عملة, فان الملف الثاني سيبقى بالانتظار لحين اكمال الملف الاول وعندها يكون بالطابور.
3. في داخل الحاسوب فان الاجهزة المختلفة لها سرع انجاز مختلفة, مثلا اجهزة الادخال بطيئة نسبة لاجهزة المعالجة لذلك فان عملية ارسال واستلام البيانات بين هذة الاجهزة يؤدي الى توقف بعض الاجهزة السريعة لانتظار البيانات من الاجهزة البطيئة, في مثل هذة الحالة فان هناك عدة طرق لمعالجة الفرق بالسرعة احدها استخدام الطابور.

4 -انظمة مراكز اتصالات التلفون تتطلب ان تستخدم الطابور لتجعل الاشخاص ينتظرون تلبية طلباتهم وتوفير الخدمة لهم وخصوصا المجانية.

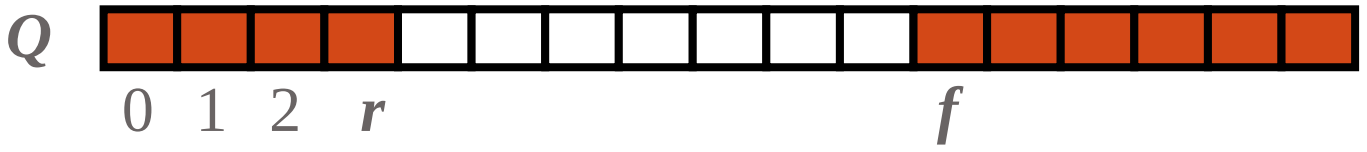
5. ال طلبات المختلفة على شبكة الانترنيت ممكن ان تكون ضمن طابور, وخصوصا في router

6. طلبات المقاطعة للبرامج داخل الحاسوب تحتاج الى استخدام الطوابير لتلبية اوامر المقاطعة المختلفة والتي تصل بوقت واحد او اوقات متقاربة.

7. انظمة التشغيل بشكل عام تستخدم الطوابير في مراحل مختلفة لتلبية الطلبات الكثيرة للمستخدم.

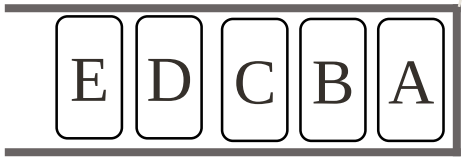
تنفيذ الرتل باستخدام مصفوفة

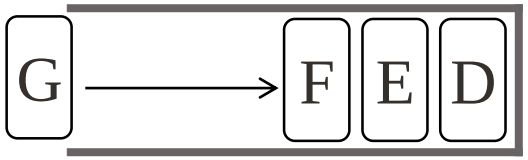
- في التمثيل باستخدام المصفوفة نستخدم دليلين indexes
  - Front لبيان موقع أول عنصر
  - Rear لبيان موقع آخر عنصر.
- عند التنفيذ باستخدام المصفوفة، نحتاج إلى العودة مجدداً إلى بداية المصفوفة، وذلك عندما يصبح Rear هو آخر دليل ممكن في المصفوفة.
- يجب أن نتأكد دوماً من أن عدد العناصر في الرتل لا يتجاوز حجم المصفوفة.

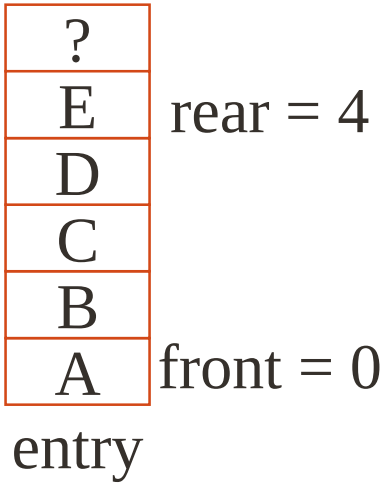


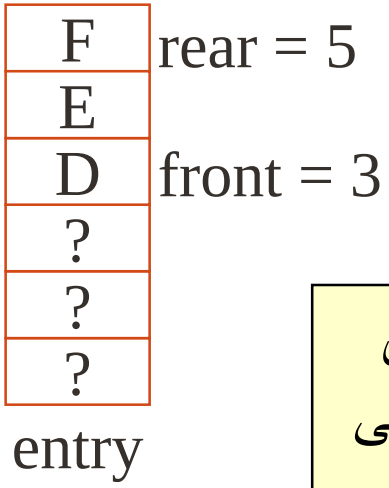
التنفيذ باستخدام المصفوفة

dequeue A,B,C; enqueue F and G..







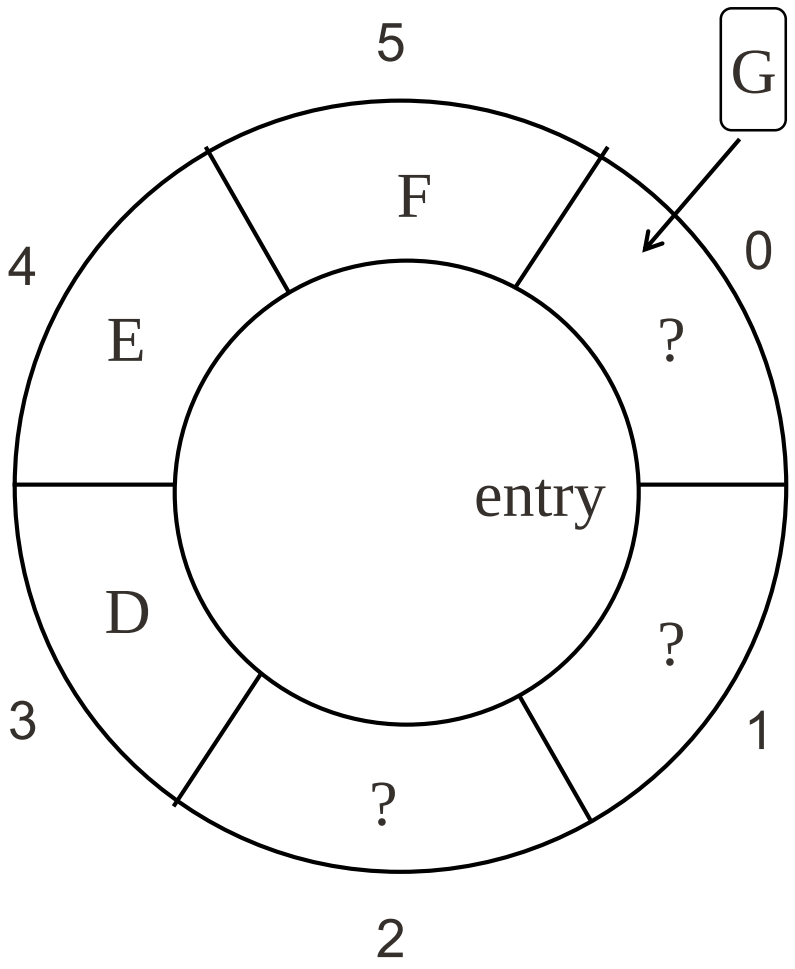


لا يمكننا إضافة ‘G’ لأن المتغير rear قد وصل إلى آخر الصفيفة.

ما الحل؟

التنفيذ باستخدام المصفوفة الدوارة Circular Array

إذا نظرنا إلى الصفيفة كحلقة، تكون الخلية التي تأتي بعد الخلية الخامسة هي الخلية رقم 0.

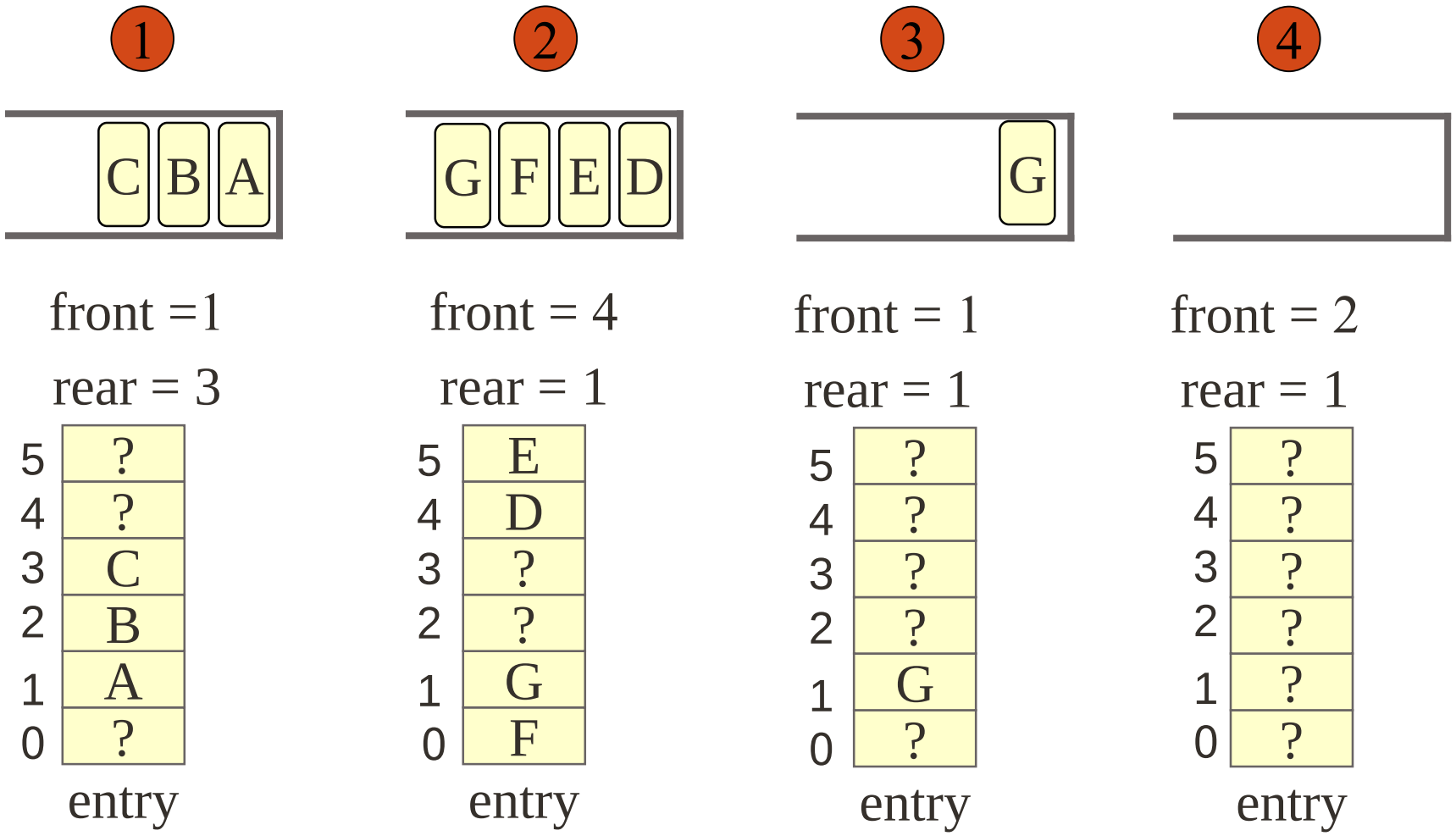


C	B	A

5	F	rear = 5 → 0
4	E	
3	D	front = 3
2	?	
1	?	
0	?	

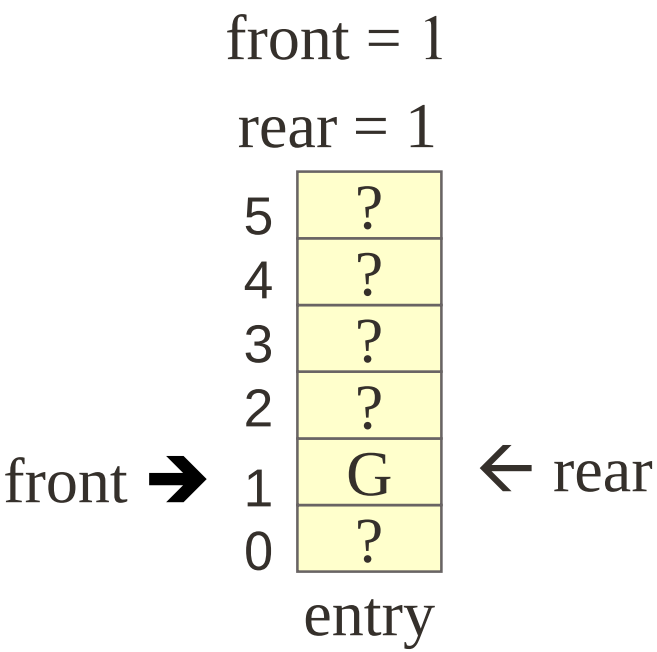
entry

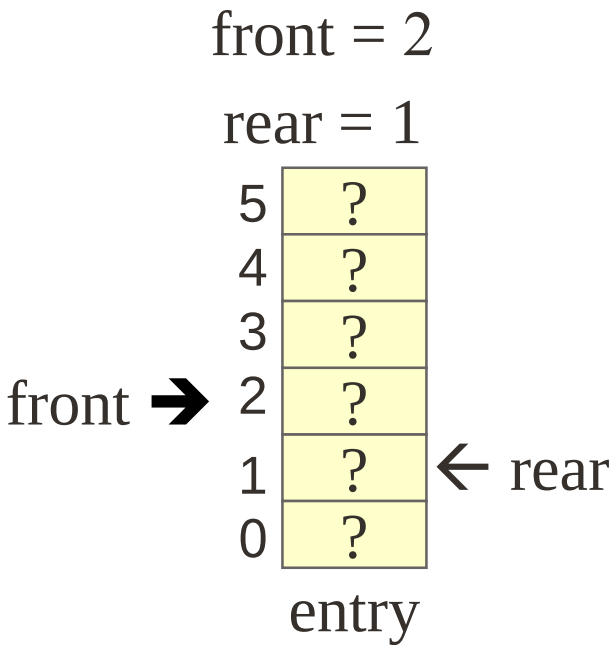
التنفيذ باستخدام المصفوفة- موقع المؤشرات



Empty ➔ (Rear== front –1)

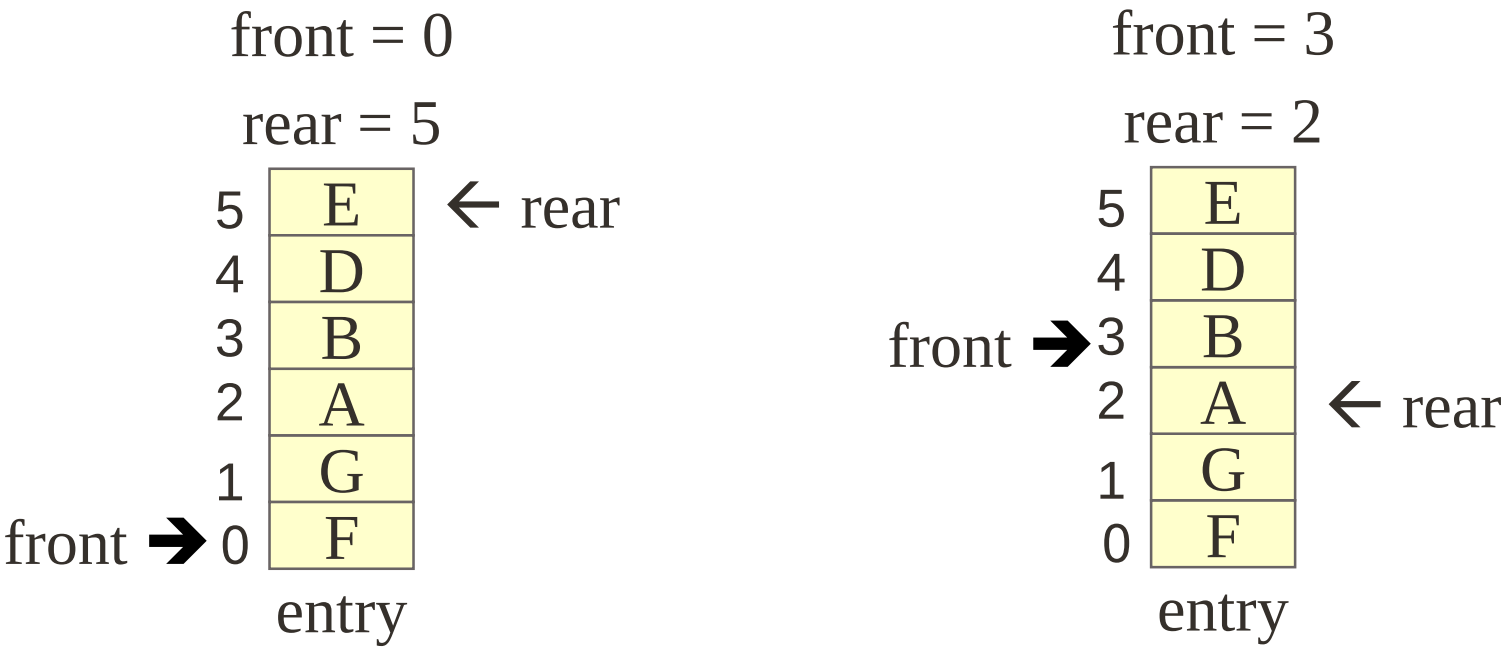
فارغة Empty ؟





front == ((rear+1))%MaxSize)

ممتلئة Full ؟



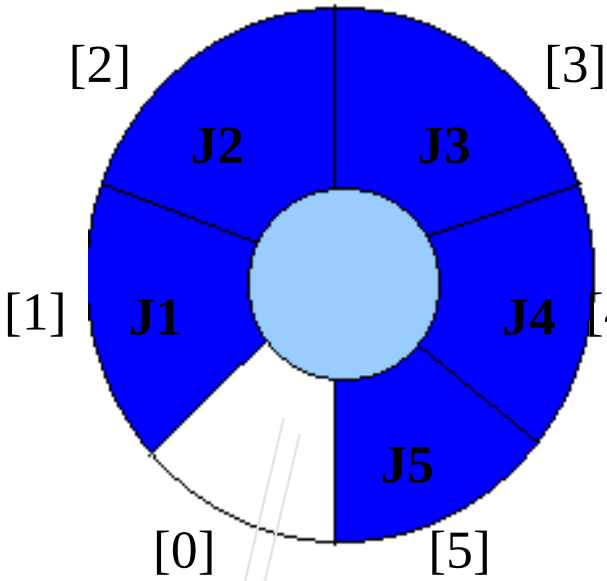
front == ((rear+1))%MaxSize)

فارغة Empty أم ممتلئة Full ؟

- رتل فارغ Empty
  - front = back + 1...
- رتل مليء Full
  - التعبير ذاته!
  - السبب : n قيمة لتمثيل n+1 حالة
- حلول
  - نجعل الصفيفة بحجم n+1 ونسمح فقط بتخزين n عنصر فقط
  - نستخدم متغير بولياني للقول صراحة أن الرتل فارغ أم لا
  - نستخدم عداد counter نخزن فيه عدد العناصر في الرتل

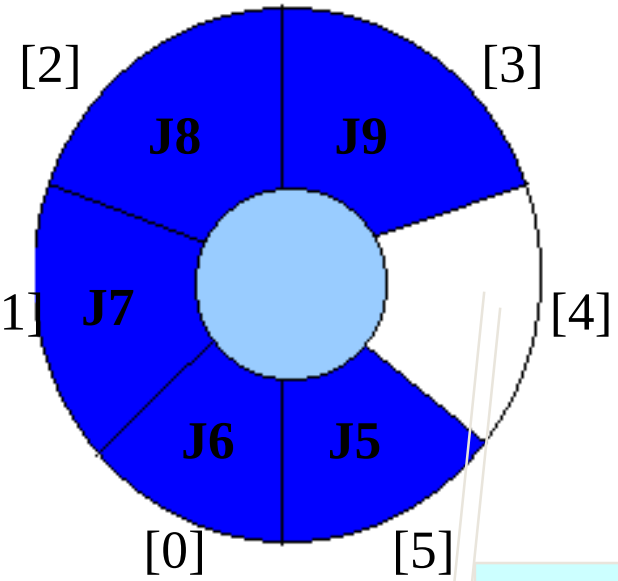
الطريقة الأولى (مصفوفة من n+1 عنصر)

FULL QUEUE



front =0
rear = 5

FULL QUEUE



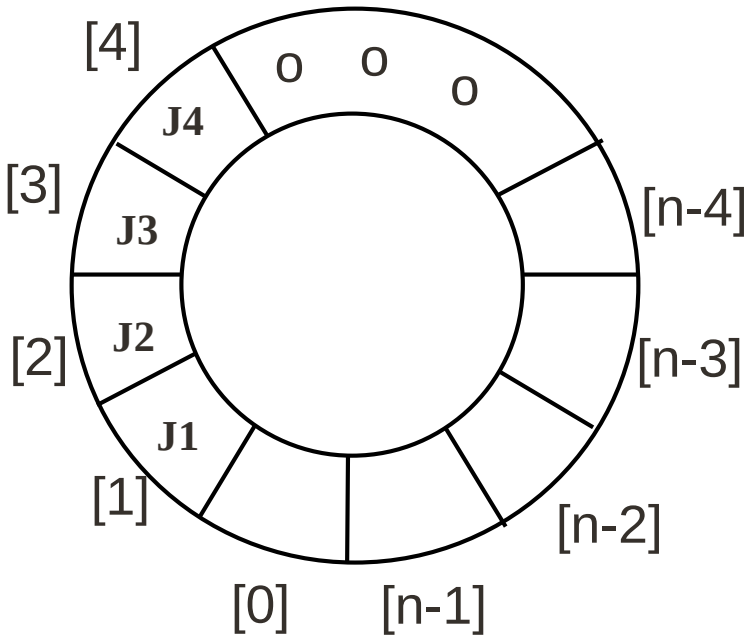
front =4
rear =3

نترك خانة فارغة عند ملء الرتل

يشير المتغير front دائماً إلى الموقع الذي يسبق أول عنصر في اللائحة

كيف نختبر كون الرتل فارغاً؟
كيف نختبر كون الرتل مليئاً؟

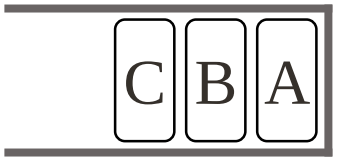
Print_Queue

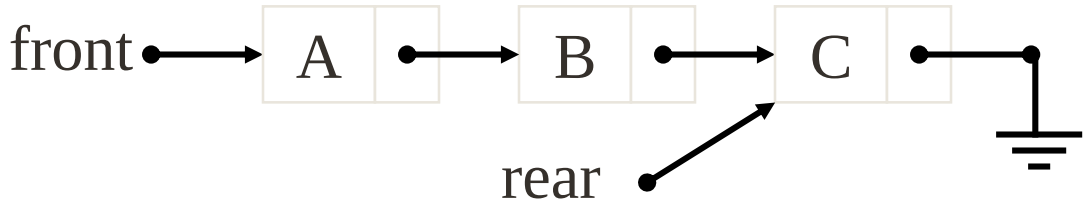


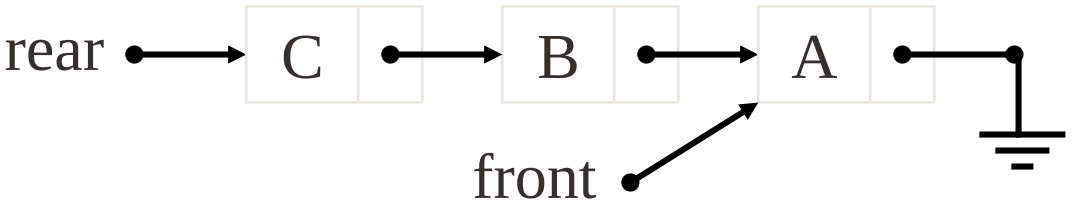
front = 0
rear = 4

Print_Queue → (J1, J2, J3, J4)

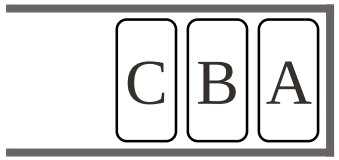
التنفيذ باستخدام لائحة مترابطة

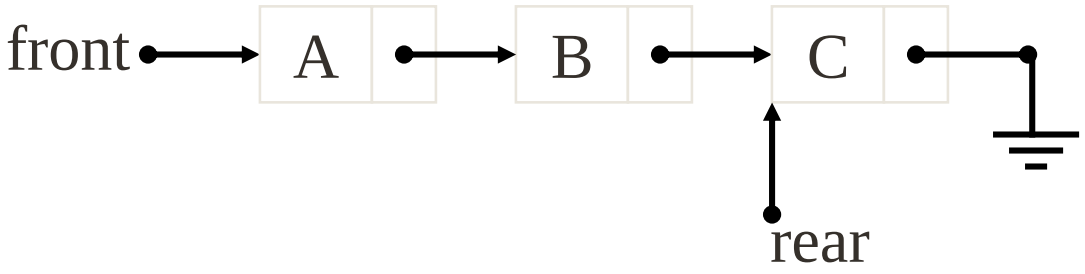






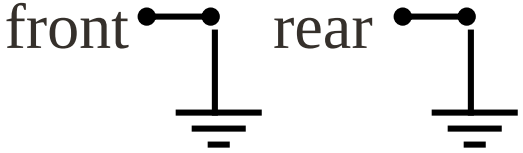
التنفيذ باستخدام لائحة مترابطة



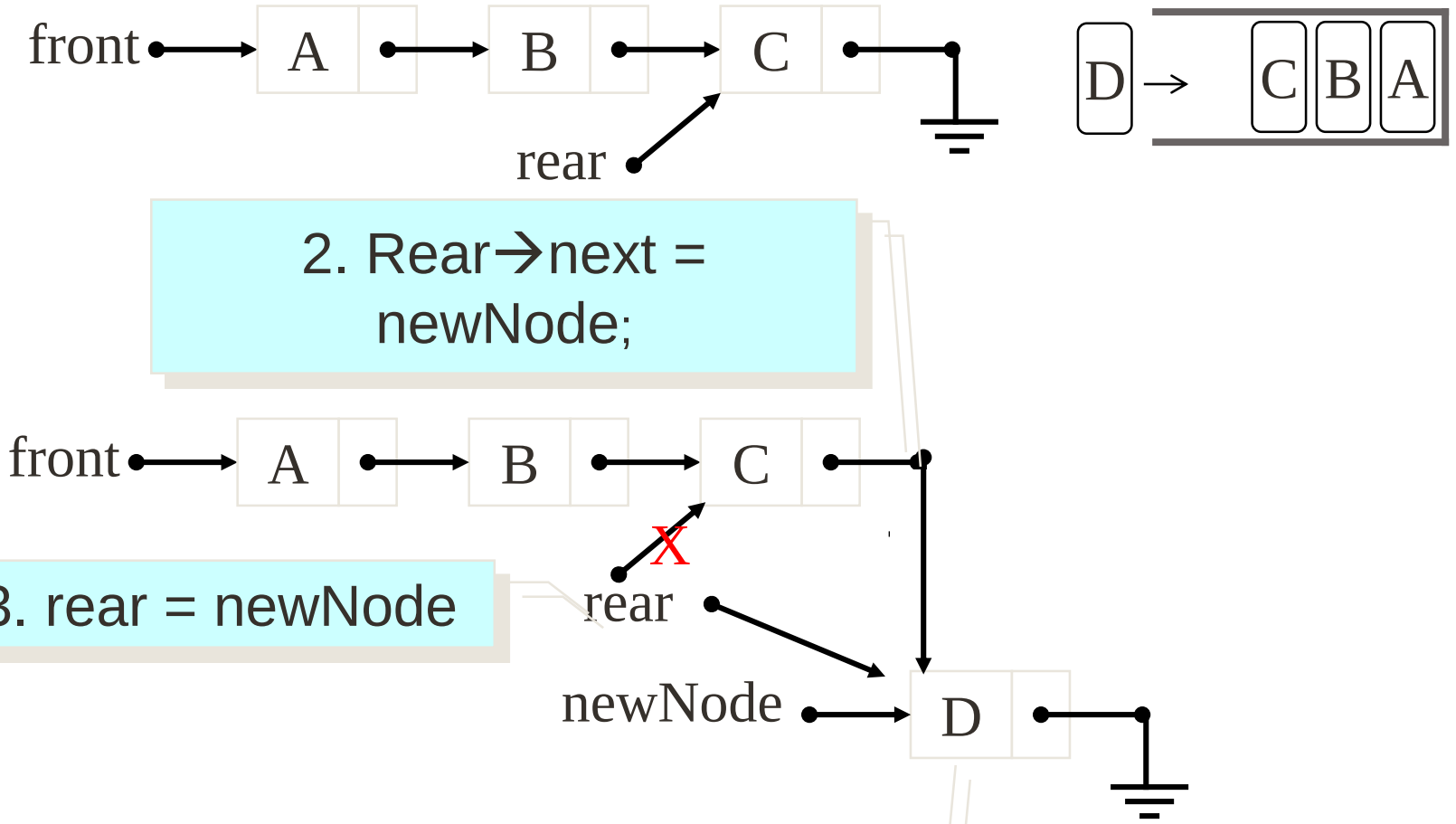


اللائحة المترابطة - إنشاء الرتل

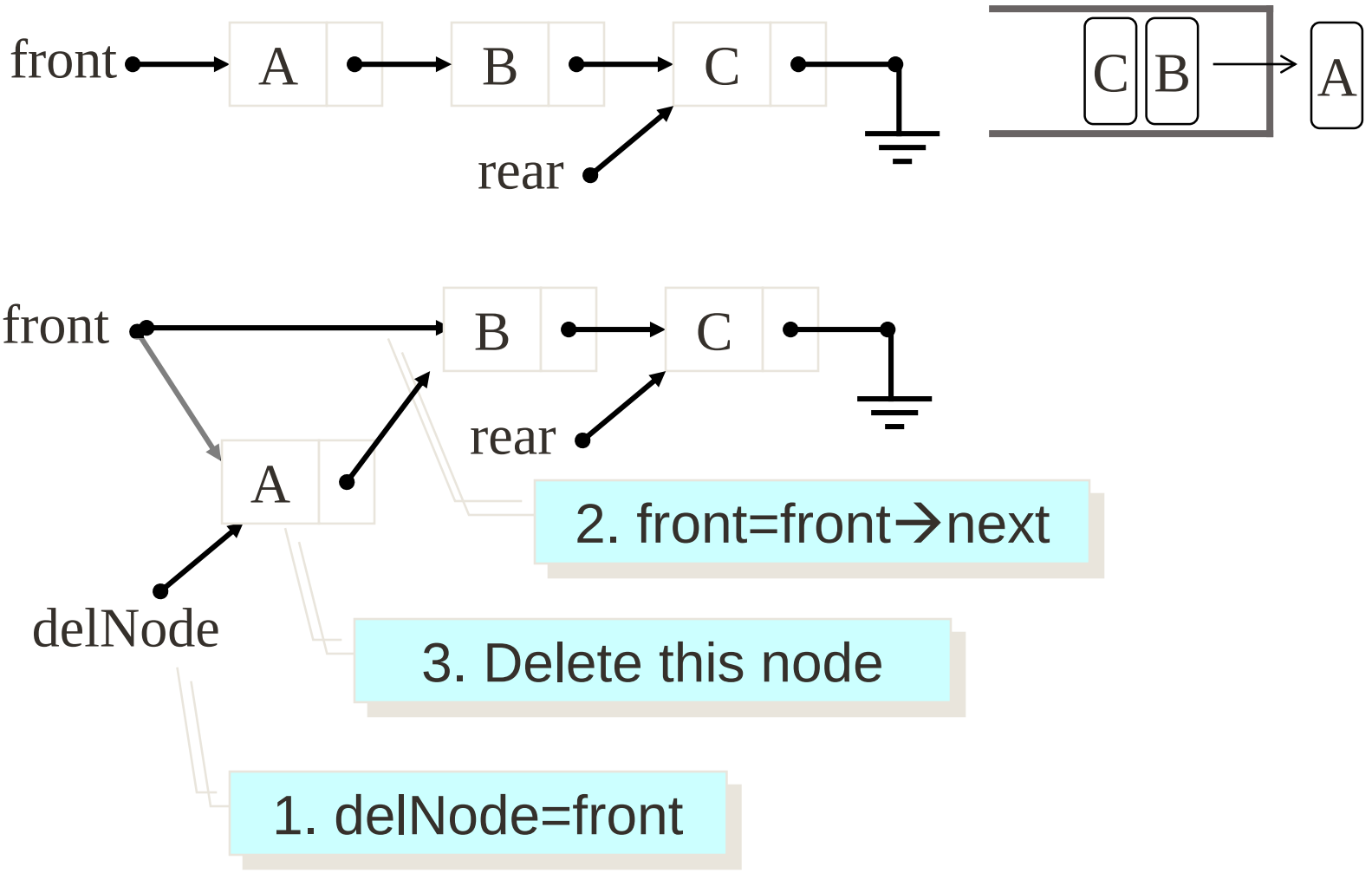
Front ← null;
Rear ← null;



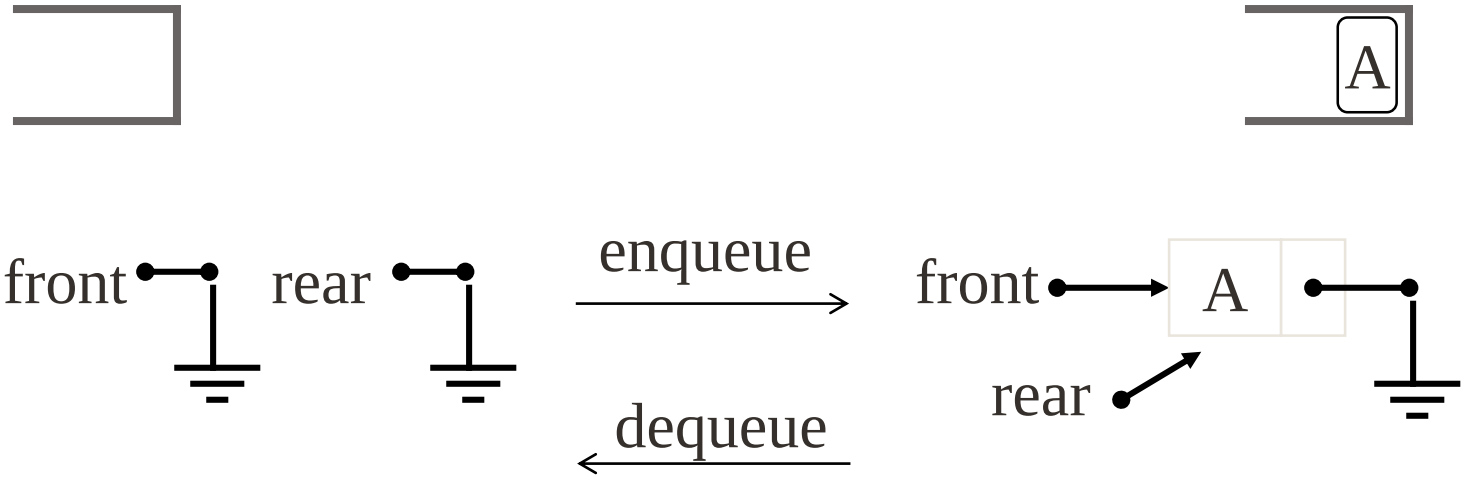
Enqueue - اللائحة المترابطة



Dequeue - اللائحة المترابطة

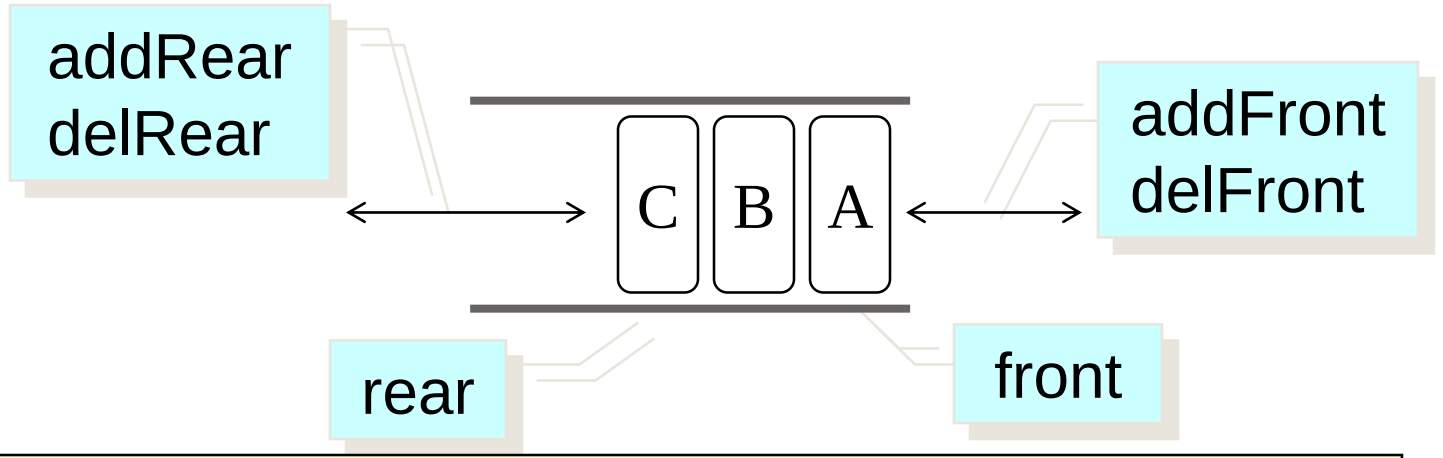


معالجة الرتل الفارغ



يجب معالجة الأرتال الفارغة خاصة في إجرائيتي Enqueue و Dequeue!

الرتل ذو النهايتين Deque



- الرتل ذو النهايتين double-ended queue مفتوح من نهايتيه.
- يمكن إضافة/حذف العناصر (addRear/ delRear) عند آخر الرتل rear، أو إضافة/حذف (addFront/ delFront) عند البداية front.
- ليس بالإمكان إضافة/حذف العناصر في منتصف الرتل ذو النهايتين.
- يمكن اعتبار المكدس والرتل حالتين خاصتين من الرتل ذي النهايتين.

يُنجز Deque باستخدام اللوائح المضاعفة الارتباط.

الرتل ذو الأولوية Priority Queue

- نمط تخزين معطيات يسمح بالإدراج بأي ترتيب كان وبالحذف وفق ترتيب مفتاح key يمثل أولوية الأغراض داخل الرتل.
- نخرج دائماً الغرض (أو أحد الأغراض) ذي الأولوية العليا.
- يجب أن ينطبق على المفاتيح علاقة ترتيب كاملة (مثل ≥ أو≤).

تطبيقات الرتل ذو الأولوية

- يُستخدم هذا الرتل عندما يكون اختيار العنصر التالي للمعالجة وفق معايير مختلفة عن معيار زمن دخول العنصر إلى الرتل.

- مثال: البحث بالعرض داخل بنى مثل البيان أو الشجرات.