

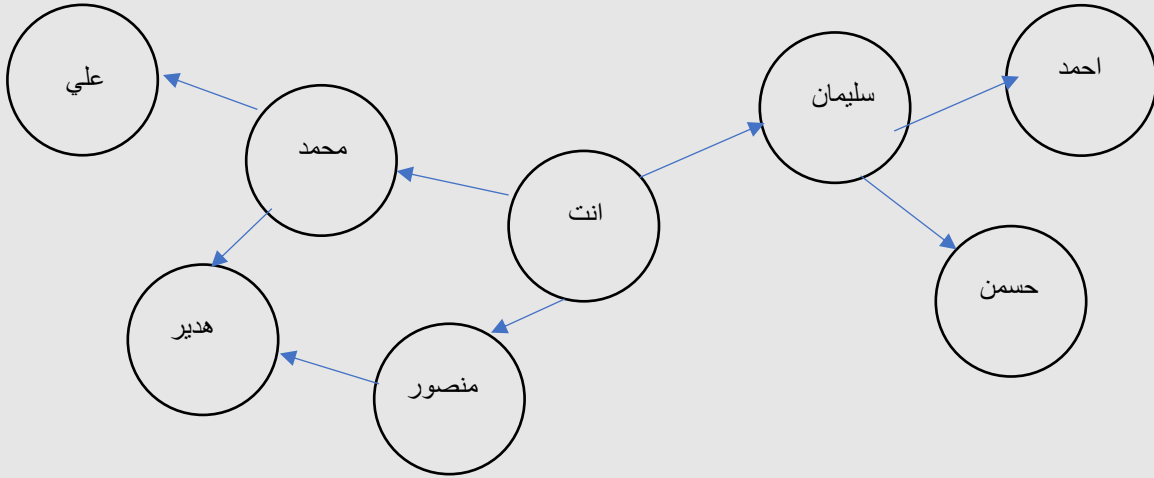
الباب السادس

الgraph

دي طريقة بنمثل بيها العلاقات يعني مثلا لو انت بتلعب كوتشينة فا ترتيب الدور بينك وبين باقي اللاعبين ممكن يتمثل بgraph اللاعب الأول -> اللاعب الثاني -> اللاعب الثالث وهكذا اللاعبين هنقول عليهم نقط او (nodes) و العلاقة الي بتربطهم سهم او edge و النقطة الواحدة ممكن تتوصل باكثر من نقطة تانية مش لازم نقطة واحدة بس و النقط الي بتتوصل ببعض بنقول عليهم جيران و ممكن النقط تبقى متوصلة ببعض بس مش بشكل مباشر يعني مثلا اللاعب الأول مفيش سهم بينه وبين اللاعب الثالث بس انا ممكن أوصل من اللاعب الأول لحد اللاعب الثالث بس هعدي على اللاعب الثاني الأول

Breadth-first search

معناها الحرفي البحث بالاتساع أولا و ده عشان انت بتدور في الحاجات الي في نفس المستوى الأول وهنفهم ده اكثر في الشرح دلوقتي انت عديت على خوارزميات بحث قبل كده زي ال binary search بس ال breadth-first دي خوارزمية بتعامل بس مع ال graph وبتجاوبلي على سؤالين 1- هل في طريق من النقطة 1 للنقطة 2- ايه هو أقصر طريق بين النقطتين



انت دلوقتي صاحب مزرعة مانجا وعايز تلاقي بياع ياخذ المحصول بتاعك وبيبيعه فهل انت تعرف بياع؟ عشان تجاوب على السؤال ده هتبص الأول على جيرانك لو فيهم بياع خلاص كده تمام انت خلصت طب لو انت خلصت كل جرانك وملقتش فيهم بياع هتبدأ تشوف جيران جيرانك وهو ده معنى breadth-first انا هبص في الحاجات القريبة مني الأول ومش هروح ابعد غير لما أخلص كل الحاجات اللي على نفس القرب مني فبعد ما تخلص جيران جيرانك لو لسه ملقتش بياع هتدور في جيران جيران جيرانك وهكذا طب انت بتعمل كده ازاى بتحط كل جرانك في list ولما بتشوف واح منهم وتلاقيه مش بياع بتشيله من ال list وتحط جرانته وهنا وبتوقف لما تلاقي بياع

أقصر طريق

انا دلوقتي بقدر اعرف إذا كانت النقطة موجودة في الgraph بتاعي ولا دلوقتي جه وقت السؤال الثاني ايه هو أقصر طريق للنقطة دي؟

انت هتفضل شخص من الدرجة الأولى على شخص من الدرجة الثانية وشخص من الدرجة الثالثة (الدرجة الأولى دول جيرانك المباشرين والدرجة الثانية جيران جيرانك وهكذا) انت مش هتدور في الدرجة الثانية غير لما تخلص الدرجة الأولى و لحسن الحظ دي الطريقة الي ال breadth-first شغال بيها انت كده كده بتضيف الناس الجديدة لآخر ال list بس ده معناه انك لازم تاخذ بالك وتدور في النقط بنفس الترتيب الي انت حطيتها بيه عشان لو غيرت الترتيب فا كده انت مش هتلاقي الطريق الأقصر و في طريقة هتأكد انك دايمًا بتدور بالترتيب اليس انت دخلت بيه النقط

الصفوف queues

الصفوف زي ما اسمها بيقتراح بتشتغل على ان أي عملية إضافة هتبقى في اخر الصف وأي عملية حذف بتكون من اول الصف وبكده هنضمن أنك دايمًا بتدور بالترتيب الصحيح للنقط

تطبيق ال graph

ازاي هنطبق الجراف بالكود؟ الجراف علاقة بين نقطتين $a \rightarrow b$ ويتصادف أنك تعرف واحد من هياكل البيانات بيشتغل بطريقة مشابهة ال hash table وفي حالة ال graph ال hash table هيكون عبارة عن نقطة ودي هتكون المفتاح بتاعنا و array من النقط وهم الجيران بتوع النقطة الي في المفتاح هل هنا هيكون الترتيب مهم؟ الإجابة لا عشان أصلا ال hash table مش مترتب الترتيب هيفرق معاك وانت بتدور لكن وانت لسه بتبني ال graph مش هيفرق إذا النقط كانت مترتبة ولا لا

تطبيق ال breadth first

هتضيف كل النقط الي من الدرجة الأولى للصف بتاعنا وتعدي عليه نقط نقطة تشوف هل فيهم بياع ولا لا لو النقطة كانت بياع فتمام هنقف وكده انت وصلت للنقطة الي انت عايزها لو مش بياع هتخزفه من الصف وتضيف كل جرائنه وحتحتاج تسجل النقط الي انت دورت فيها عشان لو النقط مش متسجلة ممكن نقطة تدخل الصف اكرر من مرة وبكده ممكن تدخل infinite loop

سرعة الخوارزمية

عشان تدور في ال graph بتاعك كله هتحتاج تعدي على كل الخطوط الي بتوصل النقط فا ده $O(\text{عدد الخطوط})$ و انك تضيف شخص للصف ده بياخد $O(1)$ بس انت عندك عدد متغير من الأشخاص الي هو عدد النقط يبقى $O(\text{عدد النقط})$ يبقى الإجمالي هيكون $O(\text{عدد النقط} + \text{عدد الخطوط})$

ملحوظة: