Comment vous préparer à un entretien chez Google

Ces trucs et astuces pour vous préparer à un entretien chez Google nous ont étés envoyés par une de nos étudiantes qui a eu un entretien avec eux. Le format de l'entretien est abordé, ainsi que ce a quoi vous devez vous attendre. A la fin de ce document, il y a des liens qui seraient bien d'aller regarder pour vous aider a preparer et planifier un entretien chez Google.

Documents à posséder avant de planifier votre entretien téléphonique :

• Une copie non officielle de votre transcription
• AU MOINS 3 jours et heures dans lesquelles vous serez disponible pour l'entretien téléphonique technique y compris le fuseau horaire dans le format suivant (ex : 6/12 – 10h – 14h UTC)
• Le langage de programmation que vous voudriez utiliser pour l'entretien : Java, C++, C ou Python
• Le numéro de telephone avec lequel l'ingénieur pourra vous contacter le plus facilement le jour de l'interview

Conseils pour l'entretien Google :

Voici quelques conseils et astuces pour vous aider à vous préparer a réussir ! Ceux qui ont étudié ce document ont tendance à BEAUCOUP mieux réussir leurs entretiens !

1) Planifier a l'avance :

Les ingénieurs de chez Google qui vous recevront en entretien n'auront que peu de temps a vous consacrer entre leurs projets, donc s'il vous plaît, réservez du temps dans votre emploi du temps aussi ! Cela vous aide aussi a vous détendre et a être plus performant durant l'entretien. Merci d'avoir du papier et un style au cas ou l'on vous demande d'écrire certaines choses.

2) A quoi s'attendre :

Jusqu'à 45 minutes d'entretien technique avec un ingénieur logiciel Google.
L'interviewer sera intéressé par votre connaissance des principes de l'informatique (données structures, algorithmes, etc.) et comment ils peuvent être utilisés dans vos solutions.

3) Questions d'entretien :

Les sujets d'entretien peuvent couvrir n'importe quoi sur votre CV (surtout si vous avez déclaré que vous êtes un expert!), des questions de codage de tableau blanc, la construction et le développement d'algorithmes complexes et l'analyse de leurs caractéristiques de performance, leurs problèmes logiques, la conception de systèmes et l'informatique de base – tables de hachage, piles, tableaux, etc. Les bases de l'informatique sont pré-requises pour tous les rôles d'ingénierie chez Google, indépendamment de l'ancienneté, en raison de la complexité et de l'échelle mondiale des projets auxquels vous participeriez.

4) Comment réussir :

Chez Google, nous croyons a la collaboration et au partage d'idées. Plus important encore, vous aurez besoin de plus d'information sur la personne qui vous fait passer l'entretien pour l'analyser et répondre à la question dans toute son étendue.
* Il est bon d'interroger votre recruteur
* Lorsqu'on vous demande fournir une solution, définissez et encadrez le problème comme vous le voyez.
* Si vous ne comprenez pas – Demandez de l'aide ou des clarifications.
* Si vous devez supposer quelque chose - vérifiez verbalement que c'est une supposition correcte !
* Décrivez comment vous voulez résoudre chaque partie de la question.
* Laissez toujours votre interviewer savoir ce que vous pensez car il / elle sera aussi intéressé par votre processus de pensée comme votre solution.
En outre, si vous êtes coincé, ils peuvent fournir des indices s'ils savent ce que vous faites.
* Finalement, écoutez – ne manquez pas un indice si vous interlocuteur essaie de vous aider !

5) Que recherche Google ?:

"Nous ne cherchons pas simplement des ingénieurs pour résoudre les problèmes dont ils connaissent déjà les réponses; nous sommes intéressés par les ingénieurs qui peuvent trouver les réponses aux questions qu'ils ne leur sont pas venus à l'esprit avant. "

Les recruteurs examineront l'approche des questions autant que la réponse :
* Le candidat écoute-t-il attentivement et comprend-il la question ?
* Are the correct questions asked before proceeding? (important!)
* Est ce que la force brute est utilisée pour répondre un problème ? (Pas bon!)
* Les choses sont-elles supposées sans vérification préalable? (pas bon!)
* Les conseils sont-ils entendus et pris en compte?
* Le candidat est-il lent à comprendre / résoudre les problèmes? (pas bon!)
* Le candidat aime-t-il trouver des solutions multiples avant de choisir la meilleure solution?
* De nouvelles idées et méthodes de traitement d'un problème sont-elles recherchées ?
* Le candidat est-il inventif et flexible dans ses solutions et ouvert aux nouvelles idées ?
* Le questionnement peut-il évoluer vers une résolution de problèmes plus complexe ?

Google est impatient de voir du code de très haute qualité, efficace et clair sans taper des erreurs. Parce que tout ingénieurs (à tous les niveaux) collaborent à travers la base de code de Google, avec un processus d'examen efficace, il est essentiel que chaque ingénieur travaille au même niveau élevé.

6) Posez plus de questions ! :

Assurez-vous d'avoir une bonne compréhension de Google en tant qu'entreprise – plus loin que les produits principaux de Google – découvrez ce que nous faisons ici :
http://www.google.com/corporate/ ou:
http://en.wikipedia.org/wiki/Google
A la fin de votre entretien, la plupart des recruteurs vont vous demandez si vous avez des questions a propos de l'entreprise, de l'environnement de travail, leurs expériences, etc. C'est intelligent d'avoir quelques questions préparées pour chaque entretien, ne vous inquietez pas si votre esprit se vide.

Si vous avez des questions sur le déroulement de l'entretien, la rémunération ou votre travail, s'il vous plaît dirigez-les vers votre recruteur.

7) En lire plus :

* Pour comprendre le fonctionnement des équipes de développement de Google
http://en.wikipedia.org/wiki/Agile_development
* Pour en savoir plus sur les principaux projets de Google -
http://labs.google.com/why-google.html
Si vous n'avez pas encore lu les conseils techniques de Steve Yegge, veuillez consulter son blog: http://steve-yegge.blogspot.com/2008/03/get-that-job-at-google.html
* En raison de la taille des produits que vous construisez, il est impératif que vous soyez à l'aise avec un gros O notation, voici où rafraîchir: http://en.wikipedia.org/wiki/Big_o_notation

8) Conseils de préparation technique ! :

Les principaux ingénieurs logiciels devraient se préparer à réussir leur entretien chez Google:

Complexité de l'algorithme: Il est assez important de comprendre l'analyse de complexité big-O.

Exécuter encore quelques problèmes pratiques pour l'obtenir dans l'application.

Tri: Savoir trier. Ne fais pas de bulles. Vous devriez connaître les détails d'au moins un n*log (n) algorithme de tri, de préférence deux (disons tri rapide et fusion). Fusionner le tri peut être très utile dans les situations où le tri rapide est impraticable, alors jetez un coup d'oeil.

Hashtables: Sans doute la structure de données la plus importante connue de l'humanité. Vous devez absolument savoir comment ils fonctionnent. Etre capable d'en implémenter un en n'utilisant que des tableaux dans votre langue favori, dans à peu près l'espace d'une interview.

Arbres: Connaître les arbres; construction d'arbre de base, algorithmes de traversée et de manipulation. Familiarisez-vous avec les arbres binaires, les arbres n-ary et les arbres-trie. Se familiariser avec au moins un type d'arbre binaire équilibré, qu'il s'agisse d'un arbre rouge / noir, d'un arbre splay ou d'un arbre AVL, et sachez comment c'est mis en œuvre. Comprendre les algorithmes de traversée d'arbre: BFS et DFS, et connaître la différence entre post-achat, pré-achat et précommande.

Graphiques: Les graphiques sont vraiment importants chez Google. Il y a 3 façons de représenter un graphique dans une mémoire (objets et pointeurs, matrice et liste d'adjacence); vous familiariser avec chaque représentation et ses avantages et inconvénients. Vous devriez connaître les algorithmes de base de traversée de graphe: recherche en largeur et première recherche en profondeur. Connaître leur complexité de calcul, leurs compromis, et comment les implémenter en code réel. Si vous avez une chance, essayez d'étudier sur un amateur d'algorithmes, tels que Dijkstra et A *.

Autres structures de données: Vous devriez étudier autant d'autres structures de données et algorithmes que possible. Vous devriez surtout connaître les classes les plus célèbres de problèmes NP-complete, comme le vendeur de voyage et le problème de sac à dos, et être en mesure de les reconnaître quand un recruteur vous les demande fourbement. Découvrez ce que signifie NP-complete.

Mathématiques: Certains recruteurs posent des questions de base discrètes en mathématiques. Ceci est plus répandu chez Google que chez d'autres entreprises parce que nous sommes entourés par de nombreux problèmes, les problèmes de probabilité et d'autres situations de mathématiques. Dépenser du temps avant l'entretien pour rafraîchir votre mémoire (ou vous enseigner) l'essentiel des combinatoires et probabilités. Vous devriez être familier avec les problèmes n-choose-k et leurs semblables – le plus sera le mieux.

Systèmes d'exploitation: Connaître les processus, les threads et les problèmes de simultanéité. Connaître les serrures, mutexes, sémaphores et moniteurs et comment ils fonctionnent. Connaître l'impasse et livelock et comment les éviter. Connaître les ressources dont un processus a besoin et les besoins d'un thread et comment le changement de contexte fonctionne, et comment il est initié par le système d'exploitation et sous-jacent Matériel. En savoir un peu sur la planification. Le monde évolue rapidement vers le multicœur, alors connaître les principes fondamentaux des constructions de simultanéité "moderne"

Codage: Vous devriez très bien connaître au moins un langage de programmation, et il devrait de préférence être C ++ ou Java. C # est ok aussi, car c'est assez similaire à Java. Vous devrez vous attendre à écrire du code dans certains de vos entretiens . Vous devrez connaître une bonne quantité des détails sur votre langage de programmation préféré.

9) Exemples de sujets :

Exemple de sujet de codage : construire / parcourir des structures de données, implémenter des routines système, distiller de grands ensembles de données à des valeurs uniques, transformer un ensemble de données en un autre.

Conception d'algorithmes / analyse
Exemples de sujets: analyse big-O, tri et hachage, manipulation de quantités de données extrêmement importantes. Voir aussi les sujets listés sous "Codage"

Conception du système
Exemples de sujets: ensembles de fonctionnalités, interfaces, hiérarchies de classes, conception d'un système sous certaines conditions contraintes, simplicité et robustesse, compromis.

Discussion ouverte
Exemples de sujets: les plus grands défis rencontrés, les meilleures / pires conceptions que vous ayez vues, l'analyse de la performance et optimisation, tests, idées pour améliorer les produits existants.

Vidéos utiles pour préparer un entretien avec Google:

https://www.youtube.com/watch?v=w887NIa_V9w

https://www.youtube.com/watch?v=qc1owf2-220&feature=youtu.be

https://www.youtube.com/watch?v=oWbUtlUhwa8&feature=youtu.be

Ressources utiles pour préparer un entretien/emploi avec Google:

https://www.google.com/about/careers/lifeatgoogle/hiringprocess/

http://www.google.com/about/careers/ https://www.youtube.com/user/lifeatgoogle

http://www.fastcompany.com/703050/google http://www.wired.com/2014/08/how-to-solve-crazy-open-ended-google-interview-questions/