Ryan, these 21 questions are for practice. A 45 minute technical interview might have three such questions. Questions marked with $\star$ are (inspired by) actual ones I've gotten. Questions marked with ! are especially challenging; I wouldn't expect more than one in an interview. Each question has a main version as well as a followup twist that an interviewer might ask upon satisfactory solution of the main version. It might be good to avoid reading these yourself, instead asking Chloe to draw randomly from these questions during a practice interview.

Questions of the form *determine xyz given abc* stand for *construct a program that determines xyz given abc*. Also implicit is the expectation that you explain what assumptions you make and that you offer an informal complexity analysis for each solution. For example, you may want to remark on your assumptions about the types and probability distribution of input data and on which parameters you regard as asymptotically large.

0. popularity contest. from a non-empty list of names identify the name(s) of greatest frequency. —— TWIST: identify the 10^4 most frequently occuring names (assumed unique) from a list of length 10^6.

1. $\star$ legal parentheses. check whether a given string of (s and)s is well-formed. —— TWIST: what if we permit both round and square brackets? we regard ([)] as ill-formed.

2. justify text. a paragraph is a list $(x_i : 0 \leq i < N)$ of words with natural widths w_i. we segment a paragraph into lines and slightly stretch each word. precisely, we choose indices $(i_j : 0 \leq j < J)$ of the first word in each jth line as well as a stretch amount $(\rho_i \in [-0.1, +0.1] : 0 \leq i < N)$ for each word. we then render the ith word at width $(1 + \rho_i)w_i$ so that each line fits, i.e., so that the excess $e_j = W - \sum_{l_j \leq i < l_{j+1}} (1 + \rho_i)w_i$ is non-negative. determine ρ_i, J, and l_j that minimize the *distortion* $D = \sum_j e_j^3 + \lambda \sum_i \rho_i^2$. —— TWIST: in a greedy approach, we set l_1 and $(\rho_i : l_0 \leq i < l_1)$ to minimize the distortion $e_0^3 + \lambda \sum_{l_0 \leq i < l_1} \rho_i^2$ of the first line, then repeat. is this greedy approach suboptimal?

3. $\star$ merge sort. implement merge sort. —— TWIST: compare merge sort with some alternatives.

4. missing numbers. given a length-$(n-1)$ array L containing distinct integers in the range $R = [0, n)$, determine the element of R not in L. —— TWIST: now L has length $n-2$ and misses two elements of R. determine these missing elements.

5. viola jones. imagine a 2d $H \times W$ array of `float`s, some negative. find a contiguous sub-array (specified by a corner (r, c) a shape $h \times w$ with $r + h \leq H, c + w \leq W$) whose elements have greatest sum? —— TWIST: what if we wish to maximize the *product*?

6. ! memory allocation. implement a simple memory allocator (i.e., implement `char* malloc(int nb_bytes)`, `char* free(int nb_bytes)`). use a primitive `char* sbrk(int nb_kilobytes)` to request chunks of memory from the OS. (`sbrk` stands for *space increments after program break address*.) —— TWIST: our video game makes a large number of small (≤ 8 byte) allocations (representing polygons etc) when rendering each frame, and it frees all of those allocations at once. adapt your memory allocator to this setting. speed is key.

7. palindromic substring. find a longest contiguous palindromic substring of a given string. —— TWIST: count the length-$\geq k$ contiguous palindromic substrings of a given string and for a given k.

8. distinct elements. consider a type that supports a comparator with respect to which it is totally ordered. determine whether a list of elements of this type has all elements distinct. —— TWIST: does your solution use as few comparisons as possible (asymptotically)?

9. ! $\star$ herding cows. a farmer tries to herd musical cows all onto some single hill in a finite set H of hills. each cow starts on some hill in H. the farmer may blare any s in a finite set S of songs; when this happens, all the cows on hill h move to the hill $f(s, h)$. given H, S, f determine whether the farmer may achieve her goal by blaring some sequence of songs. —— TWIST: what if the farmer's goal is to clear a specific hill $h_\star \in H$ of cows?

10. $\star$ sparse vectors. a sparse vector is a list of numbers most of which are zero. implement a space-saving sparse vector type along with a dot product. —— TWIST: a sparse matrix is defined accordingly. represent and multiply large sparse matrices.

11. **super spreader.** suppose we have the (reflexive, symmetric) 'physically meets' relation on a large set of people, represented for each person as a list of related people. *compute the relation's transitive closure.* choose output representation that uses a reasonable amount of space. —— TWIST: a relation's *size* is the maximum among the cardinalities of equivalence classes of the induced equivalence relation. determine a node that, when removed from the graph, results in a relation of least size.

12. **! bipartite graph.** the theory of markov chains references *k-periodic graphs.* an undirected graph is k-periodic if k divides the length of every cycle. determine whether a given graph is 2-periodic. —— TWIST: does there exist a $k \geq 2$ for which the given graph is k-periodic?

13. **⋆ family tree.** implement a binary-tree-of-strings type. ancestor trees are of this type. we display ancestor trees in this style: `george(katy(mike,carole),will(charlie(liz,phil),diana))`. compute the length of such a display. —— TWIST: compute two people's genetic relation given their ancestor trees. the genetic relation of siblings is $1/2$; of cousins is $1/8$; of strangers is 0; and so forth. the two trees' nodes name people, and we assume people have distinct and unique names.

14. **⋆ knight tours.** how many ways can a knight on an otherwise empty chessboard move in exactly n steps from one corner to the opposite corner? —— TWIST: consider a simplified game of chess with three black knights, one white king, and no other pieces. the knights start as close as possible to one corner; the king, to the opposite corner. this game has no rules about draws — no stalemate, no draw by repetition, no draw upon 50 reversible moves. *can black force a checkmate?*

15. **⋆ an evil company.** a csv file is a text file representing a 2D grid of entries, with rows delimited by newlines and with elements of a row delimited by commas. this csv file's rows records a person's data on some website. its columns `abcdef` list: `a` the website, the person's `b` username and `c` password on that website, and `d` their email address, `e` phone number, and `f` year-of-birth. *parse the file; do some two rows represent the same person?* use reasonable heuristics to identify the aforementioned columns and to answer the question. —— TWIST: suppose now that some rows have missing entries (represented by the string `UNK`). heuristically impute what missing entries you can.

16. **city hall.** a new hall is to be built among a square grid of streets. we label street intersections by (x, y) for $0 \leq x, y < 100$. it takes $|x - x'| + |y - y'|$ minutes to walk between intersections (x, y) and (x', y'). at (x, y) live $p(x, y)$ many residents. place the hall at an intersection so as to minimize the walking time from residence to hall, averaged uniformly over residents. —— TWIST: a rectangular park has northwest corners (a, b) and southeast corner (A, B). we may not place city hall in the park or on its boundary. what then?

17. **! task scheduling.** we employ three oxen on our farm: jesse, pat, and sandy. each t in a list T of 1000 tasks requires the simultaneous effort of n_t many oxen over a contiguous period of d_t many days. a schedule assigns each ox-day to some task. determine a schedule that completes all tasks in T by year's end, if possible. —— TWIST: remark on the computational hardness of this task as we modify its parameters.

18. **k SAT.** fix a finite set of logical atoms. is a given list of assertions, each of the form p and p $\rightarrow$ q, logically consistent? here, p, q, r stand for atoms or their negations. —— TWIST: what if we also permit a small number of assertions of the form $(p \wedge q) \rightarrow r$?

19. **! graph homology.** we're given a connected undirected graph G^0. an edge e is *crucial* if removing e disconnects the graph. count the crucial edges. —— TWIST: determine a sequence of edges to remove that results in a longest chain $G^0, \cdots, G^n$ of connected undirected graphs.

20. **! ⋆ lru cache.** write an lru cache that maps small strings (say, urls) to large strings (say, html). that is, implement a data structure that at any time stores $\leq k$ many key-value pairs; that permits querying for whether a key is present and its value is so; and that permits insertion of new key-value pair, with the least-recently inserted-or-queried pair removed if space is needed. —— TWIST: show us how you would increase your confidence that your code is correct.