טריקים:

1. מציאת k האיברים הגדולים ביותר בערך לא ממוין באורך n: $O(k \cdot n)$ ע"י order stat.
2. סכימת כל הזוגות של צמתים על בעצים מאוזן - $O(\log n)$.
3. סכימת כל האבות של צמתים על בעצים מאוזן - $O(n)$, הדרגה ל- build heap.
4. $T(n) = T(\frac{n}{4}) + T(\frac{3n}{4}) + 1$ הוא תמיד $O(n)$, לא $O(\log n)$!
5. כשמשנים את פרמטר החלוקה בפונקציית select מ-$\frac{1}{5}$ ל-3 הא לא מוקדמת, מ-5 ל-7 היא פחות יעילה. הוא גדולה מ-3.
6. במיון quicksort על מערכים הפוך איברים לגמרי, זה משמש ב- select לכל $O(n \log\log n)$ ולפעמים כל $O(n \log n)$.
7. בסרטת מקסימום/מינימום לא ניתן לומר איזה חדשיני.
8. אי אפשר למיין באמצעות Bucketsort k אם לא יודעים בהסתברויות מהו טווח המספרים!
9. אם נתון כי $f(n) = \Theta(g(n))$, לכורר כי לא אומר ש- $f(n) = \Theta(g(n))$. לדוגמה - סדרת.
10. בפלט חיפוש בינארי זה order statistic; ה- size מייצג אינדקס תת-העץ, לא גודל מלא הבנות. לפי כן המפתח-השמאל.
11. הפונקציה select מוגבלת ב-$O(n)$ הקטן מ- $\Theta(n)$.
12. select זה במקרה 5 הקוסיפיצנט: $T(n) \le T(\frac{n}{5}) + T(\frac{3n}{5} + c) + O(n)$. לינארי לכל α שלא מ-4, הוא Θ.
13. חיפוש, לדוגמה, שהוא תמיד $O(\log n)$, הרי הוא $O(\log\log n)$.
14. מיון quicksort זה select בלשהו ($\frac{1}{2}, \frac{n}{3}, \frac{n}{3}$ וכו') תמיד $O(n \log n)$ W.C.
15. בהסתכלות, כשנדברים בחתור סדר שלם בין 2 פונקציות f,g, יש לחלק בינהן כך: $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \to f(n) = o(g(n))$ / $O(g(n))$
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \to f(n) = \omega(g(n))$ / $\Omega(g(n))$
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c \to f(n) = \Theta(g(n))$
16. mergesort מתוקנת בהם מנפצות של איברים, הוא לא מקבל partition אלא חלוקה שווה ל-2. לכן quicksort ניתן לייסוד לחצי, אך לא mergesort.
17. בהנחת הקלה מ הסעיף (רק 2 איברים סמוכים), בעיין השל ב- $O(n)$ ביא בעזרת הדרגה של log - אל בין n ל- $\log n$.
18. בפלט $O(\log n)$, הבפסת א האיברים בלי ה- size הגדל ביותר לוקחת $O(k \log k)$.
19. כשתנים להשיב צמן w.c. הנמוך ביותר לי מ פלטות ומוגדר 2 מצבים, הדרך את מה ש ה- Amortize הנמוך ביותר.
20. אם מלגוריתמי select 3 בנקודות 5 הנכונה היא $T(n) \le T(\frac{n}{3}) + T(\frac{2n}{3} + c)$

יחס-שבל שהפיתו:

$f(n) = 2^{\sqrt{\log\log n}}$	o	$g(n) = \log\log(n)$
$A(n) = \log(T(n))$, $T(n) = 1(\log n) + 1$	o	$B(n) = T(\log n)$
$f(n) = \frac{n}{\log n}$	o	$g(n) = \frac{n^2}{2002 \log n}$
אין לו סדר $f(n)$ לא מתרן לאומנן	o	אין לו סדר $g(n)$ הקנים ל שרשרת מינימום

סכומים:

$$\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}$$

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

$$\sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{i=m}^{n} x^i = \frac{x^m - x^{n+1}}{1-x}$$

$$\sum_{i=1}^{n} \frac{1}{i} = \Theta(\log n)$$

טיפים לשאלות פרוצות:

1. כשנדברים לבוכים חסם עליון ותחתון לבעיה כלשהי של שיטת ההשוואה (עץ, Hash, שרשרת וכו'), חסם תחתון הוא לבוכים כשיטת "אם הינו יכולים בפחות מ- $\log n$, היינו מגיעים בפחות הנחות לחסם שיטת ההשוואות".
2. לקרוא היטב את השאלה, לנתחה בפרוטרוט עם הסיטה, ויקח את לבדוק אלגוריתמים רלוונטיים. יותר מזמן, אבל חוסך טעויות הבנת הנקרא של נקודות ויתר.
3. "הוכחת עמן ריבה" מתיחס לדיון מקיף על פעל בהתבסס בשאלות (מיון, סכומה וכו') לפי סבוכיות, ואל תשכח "סך הכל".
4. ניתן לחבר שרשרה השורה בעזרת ל-2 שרשרות תקינות, אותה שרשרת הא הבקנים והשניה שרשרת הבנים האיבר - בקנים ב- $O(n)$. מוגאים תכנון select מלגוריתם ובנו, יותר מהכל בין סדר חיפוש.
5. מבקר נתונים יעיל לשמירת נקודות במשור הוא לחיפוש בינארי שאלתן שרשרת הנקודה לקואורדינטות בינאריות.
6. כשמבקשים לסדר נתונים שנמצאים בקבוצה יקום של שרשרת, מין תמורת בצמן הקיצוני perfect Hash - לא שמירת מקום.
7. כשיש לנקודה קיים, ניתן לחפש בשרשרת מקומית דו-כיוונית. בתחילה בהנחתן מ בבוא לאיבר היא $O(n)$, והיא מאוזנת. אם מבקשים שמחפש $\log n$ לא לבמרן - לא לשמות תוצאה בינארי.

טריקים:

1. ניתן לדפדר לחיפוש בינארי בסל אוגר שחור ב- $O(n)$ ל בי בבסיס והאצה של צמתים.
2. תכונת מאוזנת של בני מלבנים ממוינים בשלבים ניתן לחפש ב- $O(\log n)$.
3. בסדרת Rotate ל לא מאוזן (לא) כמעט לא בנמצא - לא מעניין אותנו מקרה מעגלי.
4. שמבה לה המינימום בסל Avl הוא בין 0 ל-1.
5. אם משנים את Heapify לחיות בסיבוכיות $O(\log\log n)$, הסיבוכיות של build heap היא עדיין $O(n)$.
6. אם בותרים לחלק מערך ממוין בחיפוש k, בעזרת חיפוש בינארי, הסיבוכיות עדיפה מ- $O(\log n)$ ל- $O(k \log k)$.
7. $T(n) = T(\alpha(c-1))T(n) + T(n) + n$ $= O(n \log n)$ בדומה ל- $T(n) = 2T(\frac{n}{2}) + n$.
8. כשמחפשים 2 מספרים במערך לא ממוין שסכום הפיצוליות הוא בדיוק x (לא 2002 וכו'). $O(n)$ בעזרת Hash ומעבר לבסוף.
9. כשמחפשים מקום נתונים לאמצע קטעים והחיצוניים שבה xxxxxx, עדיף הכנסה, מחיקה ומציאה ב- $O(\log n)$ וכפלה 5 ב- $O(n)$ עדיף 3-2 ורשימה מקושרת לא הגיונית.
10. כשצריך לשמור אם במערך לא ממוין שני איברים שקיבלו נכסים מוסדרים הוא $\frac{n}{2}$, מגיעים חצון בחצאי ומסתכלים לפי איחוד. $O(n)$.
11. נשתדל לצמן W.C. לל בעזרת splay-tree, שליפת מפתחות תמיד הכוחן ($O(n)$ ל) אם קשור ב- m מ-l אחרונה לה לאמן.
12. בחלוקה שחור, אבחון של צמתים טרם שמתחת הם 0 או 1.
13. כל משפחה 2-אוניברסלית היא אוניברסלית, כל משפחה אוניברסלית היא פסאודו-אקראית.
14. מערך באורך A שלם שרכיבים הם בין $\{1, 2, \dots, m\}$ ניתן למיין ב- $O(\log\log n)$.
15. מערך מאוזן (בעל) m, כש-n חשוב הוא 1 ו-m גדול. למצוא את n בזמן $O(\log(\frac{n}{m}))$ (וגם קשור ל- m).
16. מיון מערך של מספרים בטווח תמורה - $O(\log n)$ כמו mergesort.

טריקים:

1. מציאת k האיברים הגדולים ביותר בערך לא ממוין באורך n: $O(k \cdot n)$ ע"י order stat.
2. סכימת כל הזוגות של צמתים על בעצים מאוזן - $O(\log n)$.
3. סכימת כל האבות של צמתים על בעצים מאוזן - $O(n)$, הדרגה ל- build heap.
4. $T(n) = T(\frac{n}{4}) + T(\frac{3n}{4}) + 1$ הוא תמיד $O(n)$, לא $O(\log n)$!
5. כשמשנים את פרמטר החלוקה בפונקציית select מ-$\frac{1}{5}$ ל-3 הא לא מוקדמת, מ-5 ל-7 היא פחות יעילה. הוא גדולה מ-3.
6. במיון quicksort על מערכים הפוך איברים לגמרי, זה משמש ב- select לכל $O(n \log\log n)$ ולפעמים כל $O(n \log n)$.
7. בסרטת מקסימום/מינימום לא ניתן לומר איזה חדשיני.
8. אי אפשר למיין באמצעות Bucketsort k אם לא יודעים בהסתברויות מהו טווח המספרים!
9. אם נתון כי $f(n) = \Theta(g(n))$, לכורר כי לא אומר ש- $f(n) = \Theta(g(n))$. לדוגמה - סדרת.
10. בפלט חיפוש בינארי זה order statistic; ה- size מייצג אינדקס תת-העץ, לא גודל מלא הבנות. לפי כן המפתח-השמאל.
11. הפונקציה select מוגבלת ב-$O(n)$ הקטן מ- $\Theta(n)$.
12. select זה במקרה 5 הקוסיפיצנט: $T(n) \le T(\frac{n}{5}) + T(\frac{3n}{5} + c) + O(n)$. לינארי לכל α שלא מ-4, הוא Θ.
13. חיפוש, לדוגמה, שהוא תמיד $O(\log n)$, הרי הוא $O(\log\log n)$.
14. מיון quicksort זה select בלשהו ($\frac{1}{2}, \frac{n}{3}, \frac{n}{3}$ וכו') תמיד $O(n \log n)$ W.C.
15. בהסתכלות, כשנדברים בחתור סדר שלם בין 2 פונקציות f,g, יש לחלק בינהן כך: $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \to f(n) = o(g(n))$ / $O(g(n))$
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \to f(n) = \omega(g(n))$ / $\Omega(g(n))$
$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c \to f(n) = \Theta(g(n))$
16. mergesort מתוקנת בהם מנפצות של איברים, הוא לא מקבל partition אלא חלוקה שווה ל-2. לכן quicksort ניתן לייסוד לחצי, אך לא mergesort.
17. בהנחת הקלה מ הסעיף (רק 2 איברים סמוכים), בעיין השל ב- $O(n)$ ביא בעזרת הדרגה של log - אל בין n ל- $\log n$.
18. בפלט $O(k)$, הבפסת א האיברים בלי ה- size הגדל ביותר לוקחת $O(k \log k)$.
19. כשתנים להשיב צמן W.C. הנמוך ביותר לי מ פלטות ומוגדר 2 מצבים, הדרך את מה ש ה- Amortize הנמוך ביותר.
20. אם מלגוריתמי select 3 בנקודות 5 הנכונה היא $T(n) \le T(\frac{n}{3}) + T(\frac{2n}{3} + c)$

21. ניתן לדפדר לחיפוש בינארי בסל אוגר שחור ב- $O(n)$ בי בבסיס והאצה של צמתים.
22. תכונת מאוזנת של בני מלבנים ממוינים בשלבים ניתן לחפש ב- $O(\log n)$.
23. בסדרת Rotate לא לל מאוזן (לא) כמעט לא בנמצא - לא מעניין אותנו מקרה מעגלי.
24. שמבה לה המינימום בסל Avl הוא בין 0 ל-1.
25. אם משנים את Heapify להיות בסיבוכיות $O(\log\log n)$, הסיבוכיות של build heap היא עדיין $O(n)$.
26. אם בותרים לחלק מערך ממוין בחיפוש k, בעזרת חיפוש בינארי, הסיבוכיות עדיפה מ- $O(\log n)$ ל- $O(k \log k)$.
27. $T(n) = T(\alpha(c-1))T(n) + T(n) + n$ $= O(n \log n)$ בדומה ל- $T(n) = 2T(\frac{n}{2}) + n$.
28. כשמחפשים 2 מספרים במערך לא ממוין שסכום הפיצוליות הוא בדיוק x (לא 2002 וכו'). $O(n)$ בעזרת Hash ומעבר לבסוף.
29. כשמחפשים מקום נתונים לאמצע קטעים והחיצוניים שבה xxxxxx, עדיף הכנסה, מחיקה ומציאה ב- $O(\log n)$ וכפלה 5 ב- $O(n)$ עדיף 3-2 ורשימה מקושרת לא הגיונית.
30. כשצריך לשמור אם במערך לא ממוין שני איברים שקיבלו נכסים מוסדרים הוא $\frac{n}{2}$, מגיעים חצון בחצאי ומסתכלים לפי איחוד. $O(n)$.
31. נשתדל לצמן W.C. לל בעזרת splay-tree, שליפת מפתחות תמיד הכוחן ($O(n)$ ל) אם קשור ב- m מ-l אחרונה לה לאמן.
32. בחלוקה שחור, אבחון של צמתים טרם שמתחת הם 0 או 1.
33. כל משפחה 2-אוניברסלית היא אוניברסלית, כל משפחה אוניברסלית היא פסאודו-אקראית.
34. מערך באורך A שלם שרכיבים הם בין $\{1, 2, \dots, m\}$ ניתן למיין ב- $O(\log\log n)$.
35. מערך מאוזן (בעל) m, כש-n חשוב הוא 1 ו-m גדול. למצוא את n בזמן $O(\log(\frac{n}{m}))$ (וגם קשור ל- m).
36. מיון מערך של מספרים בטווח תמורה - $O(\log n)$ כמו mergesort.

37. לפי select $T(n) = T(\frac{n}{2}) + O(n)$.
38. בפלטת ההקצה של הערוש ב- Delete שלו (המחוסר אור בבנייה בלאה מ- אמת-delete) לוקחת יותר בזמן בשרשרת מ- יתשאר בקלות יותר.
39. הדבה לל לה מבפלטת בשמן דרן splay לוקחת יותר בזמנים בשרשרת ביצוע מאשר בשרשרת מאוזנת בשרשרה מ- יתה.
40. המודל הטוב ביותר לתמוך ב- find, insert, delete ופעולות לל ערל הוא עץ מאוזן (אדומשחור לדוגמה) ב- SM של $O(\log n)$ לכל הפעולות.
41. פיבונאצ'י: $T(n) = T(n-1) + T(n-2)$, $T(1) = T(2) = 1$, בדלה יותר מכל n לל אקספוננציאלי, כלומר $T(n) = \omega(n^{2002})$.
42. אם נתון כי $f(n)$ אחרי לטווח סיור (ניתן בין $\omega(n)$ ל $o(n)$) אל אי אפשר לומר דבר לל יחס אסימפטוטי, אלא הוא מעקב-ל-מספר אי (אם מתלגוריתם כלשהי) אל כן ניתן לחפש יחס אדול.
43. לגמן Amortize לל פעולות בהשמה מתשרת, כשמערך insert-find ברוב לגמן לאיבר אחת בלבד, הוא $O(n)$.
44. בעזרת מון באמצעות הלוגרים בעזרת, לגמן שקלל של מספר הפעולות ניתן לגמן ב- $O(n)$ הפעולות ל' מבינות הגמנות והחלפות בא העזרת, והלוגרה רקורסיבית לל מערך של $n-1$.
45. אם מלגוריתם quicksort זה לקה ל-3 (בבלה, קטנים, גדולים) הסבלת בעזרן $O(n \log n)$.