Notes and hints sample field test:

Q3. You can eyeball the graph and figure out the total number of comparisons for all nodes at each level and average them up.

Regular Expressions:

A regular expression, often called a pattern, specifies a set of strings required for a particular purpose. A simple way to specify a finite set of strings is to list its elements or members. However, there are often more concise ways: for example, the set containing the three strings "Handel", "Händel", and "Haendel" can be specified by the pattern H(ä|ae?)ndel; we say that this pattern matches each of the three strings. However, there can be many ways to write a regular expression for the same set of strings: for example, (Hän|Han|Haen)del also specifies the same set of three strings in this example.

Most formalisms provide the following operations to construct regular expressions.

Boolean "or"

A vertical bar separates alternatives. For example, gray|grey can match "gray" or "grey".

Grouping

Parentheses are used to define the scope and precedence of the operators (among other uses). For example, gray|grey and gr(a|e)y are equivalent patterns which both describe the set of "gray" or "grey".

Quantification

A quantifier after an element (such as a token, character, or group) specifies how many times the preceding element is allowed to repeat. The most common quantifiers are the question mark ?, the asterisk \* (derived from the Kleene star), and the plus sign + (Kleene plus).

[26]

? The question mark indicates zero or one occurrences of the preceding element. For example, colou?r matches both "color" and "colour".

\* The asterisk indicates zero or more occurrences of the preceding element. For example, ab\*c matches "ac", "abc", "abbc", "abbbc", and so on.

+ The plus sign indicates one or more occurrences of the preceding element. For example, ab+c matches "abc", "abbc", "abbbc", and so on, but not "ac".

The preceding item is matched exactly n times.

{min,}[26] The preceding item is matched min or more times.

{,max}[26] The preceding item is matched up to max times.

{min,max}[26] The preceding item is matched at least min times, but not more than max times.

Wildcard

The wildcard . matches any character. For example, a.b matches any string that contains an "a", and then any character and then "b"; and a.\*b matches any string that contains an "a", and then the character "b" at some later point.

These constructions can be combined to form arbitrarily complex expressions, much like one can construct arithmetical expressions from numbers and the operations +, −, ×, and ÷.

The precise syntax for regular expressions varies among tools and with context; more detail is given in § Syntax.

To avoid parentheses it is assumed that the Kleene star has the highest priority, then concatenation and then alternation. If there is no ambiguity then parentheses may be omitted. For example, (ab)c can be written as abc, and a|(b(c\*)) can be written as a|bc\*. Many textbooks use the symbols ∪, +, or ∨ for alternation instead of the vertical bar.

**Examples:**

* a|b\* denotes {ε, "a", "b", "bb", "bbb", ...}
* (a|b)\* denotes the set of all strings with no symbols other than "a" and "b", including the empty string: {ε, "a", "b", "aa", "ab", "ba", "bb", "aaa", ...}
* ab\*(c|ε) denotes the set of strings starting with "a", then zero or more "b"s and finally optionally a "c": {"a", "ac", "ab", "abc", "abb", "abbc", ...}
* (0|(1(01\*0)\*1))\* denotes the set of binary numbers that are multiples of 3: { ε, "0", "00", "11", "000", "011", "110", "0000", "0011", "0110", "1001", "1100", "1111", "00000", ... }