Assignment 5

Bookmark this page

Assignment 5

Due date: Wednesday, February 16th, 5:00 pm Eastern Time

Coverage: Module 06

Files to submit: a05q1.rkt, a05q2.rkt, a05q3.rkt

Assignment Guidelines

Language Level

- Beginning Student with List Abbreviations

General Expectations

- Remember to use defined constants and helper functions where appropriate.
- Follow the full design recipe and the Coding Style document up to and including Section 3.6.2.

Allowable Racket Functions and Special Forms

- `define`, `and`, `or`, `not`, `cond`, `else`, `check-expect`, `check-within`, `equal?`
- Any type predicate such as `number?` or `boolean?`
- Any functions on numbers found in Section 2.6 of the Racket documentation
- Any functions on characters found in Section 2.11 of the Racket documentation
- Any functions on strings found in Section 2.12 of the Racket documentation
- The function `symbol=?`
- The list functions and values `cons`, `first`, `second`, `third`, `rest`, `empty`, `empty?`, and `list`
- You must **not** use `boolean=?`

Other Information

- All test data for correctness will always meet the stated assumptions for consumed values.

Question 1: Prime Factor Decomposition

A positive integer can be written uniquely as a product of prime factors. We call this *prime factor decomposition*. For example, $24 = 2^3 \cdot 3$, and $42 = 2 \cdot 3 \cdot 7$.

We will create a new data type, a special kind of list, to represent this kind of data:

```
;; A PFD, or prime factor decomposition, is a (listof Nat)
;; Requires: the elements are in ascending order
;;           the elements are prime numbers.
```

Include this data definition in your file.

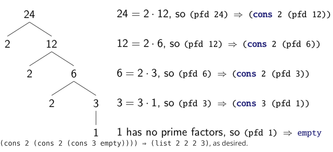
Instead of storing exponents, we will just list the prime factors, with repetition. So $24 = 2^3 \cdot 3 = 2 \cdot 2 \cdot 2 \cdot 3$ will be represented as `(list 2 2 2 3)`, and $42 = 2 \cdot 3 \cdot 7$ will be represented as `(list 2 3 7)`. The number 1 is the result of the Empty Product, so its prime factor decomposition is `empty`. A prime number has only one prime factor, so the PFD of a prime will be a list that contains that value only.

Write a function `pfd`. It consumes a positive integer *n*, and produces the PFD that represents *n*. For example,

```
> (pfd 12)
(list 2 2 3)
> (pfd 1)
empty
> (pfd 8191) ; a Mersenne prime:
(list 8191)
```

You can generate the solution list in the correct order using an appropriate recursion on natural numbers. It should not be necessary to sort.

Hint: Imagine doing prime factor decomposition by hand, then use this to inform your code design. I might calculate the PFD of a number by drawing a tree. At each node, I find the smallest prime factor of the number I'm considering, and divide it out:



$24 = 2 \cdot 12$, so `(pfd 24)` ⇒ `(cons 2 (pfd 12))`

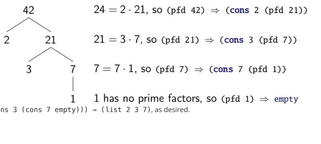
$12 = 2 \cdot 6$, so `(pfd 12)` ⇒ `(cons 2 (pfd 6))`

$6 = 2 \cdot 3$, so `(pfd 6)` ⇒ `(cons 2 (pfd 3))`

$3 = 3 \cdot 1$, so `(pfd 3)` ⇒ `(cons 3 (pfd 1))`

1 has no prime factors, so `(pfd 1)` ⇒ `empty`

So it must be that `(pfd 24)` → `(cons 2 (cons 2 (cons 2 (cons 3 empty))))` → `(list 2 2 2 3)`, as desired.



$42 = 2 \cdot 21$, so `(pfd 42)` ⇒ `(cons 2 (pfd 21))`

$21 = 3 \cdot 7$, so `(pfd 21)` ⇒ `(cons 3 (pfd 7))`

$7 = 7 \cdot 1$, so `(pfd 7)` ⇒ `(cons 7 (pfd 1))`

1 has no prime factors, so `(pfd 1)` ⇒ `empty`

So it must be that `(pfd 42)` → `(cons 2 (cons 3 (cons 7 empty)))` → `(list 2 3 7)`, as desired.

A few things to consider:

1. The base case is strange; what is it?
2. We are not just counting up or down; how are we approaching the base case?
3. Somehow hidden we need to count up to track which prime divisor we are considering.

Submit your solution in the file `a05q1.rkt`.

Question 2: Scrabble Sorting

Sometimes when sorting, we want to sort using a property of each value, instead of the value itself. Then we have to consider what to do for values for which the property is equal.

For example, we could sort a list of strings in ascending order by length. Then `(list "able" "was" "I" "ere" "I" "saw" "Elba")` could become `(list "I" "I" "saw" "ere" "was" "Elba" "able")`, but it could just as well become `(list "I" "I" "was" "ere" "saw" "able" "Elba")`, or several other options. These lists are in increasing order by length. Which should we prefer?

It turns out that one approach that is often useful is to make the sorting *stable*. A method of sorting is *stable* if after sorting, the order of items that are "the same" according to the sorting criterion is unchanged.

For example, `(list "I" "I" "was" "ere" "saw" "able" "Elba")` is the result of sorting `(list "able" "was" "I" "ere" "I" "saw" "Elba")` by string length, using a stable sort. On the other hand, `(list "I" "I" "saw" "ere" "was" "Elba" "able")` has not been sorted in a stable manner, since `"saw"` and `"was"` have the same length, but `"saw"` previously came after `"was"`, and now comes before.

In Assignment 4, you wrote a function to calculate the Scrabble score of a single word (using an archaic Dutch scoring table). We will use this to determine how to sort values.

Write a function `(scrabble-sort words)`. It consumes a list of strings, where each string contains only uppercase letters A-Z. The function produces a list containing the same strings, sorted in increasing order according to their Scrabble scores. The sort shall be *stable*.

For example, here are the Scrabble scores of a few words:

- **IT:** 3
- **OF:** 5
- **CAN:** 7
- **DEAN:** 5
- **JEST:** 9
- **ONE:** 3

Consider this example. It is a stable sort by Scrabble score, as required.

```
> (scrabble-sort (list "IT" "OF" "CAN" "DEAN" "JEST" "ONE"))
(list "IT" "ONE" "OF" "DEAN" "CAN" "JEST")
```

Sorting puts them in order by score. But since IT came before ONE, and they have equal scores, IT still comes before ONE. Similarly, OF and DEAN have equal scores, and remain in the same order.

Think about modelling your solution upon the insertion sort algorithm shown in Lesson 06.4, and writing your own helper function to compare two words. If a word appears multiple times in the input list, it should appear the same number of times in the result.

You may use your own solution to the Scrabble scoring problem from Assignment 4. If you prefer, you may use the posted solution, once it is available.

Submit your solution in the file `a05q2.rkt`.

Question 3: Books

We are going to think about lists of length exactly 3, representing an author, title, and number of pages. Each such list we will call a **Book**. Because **Book** is a compound, we'll include a constructor and and selectors for it:

```
;; a Book is a (list Str Str Nat)

;; Constructor
(define (make-book author title pages) (list author title pages))
;; Selectors
(define (book-author b) (first b))
(define (book-title b) (second b))
(define (book-pages b) (third b))

;; A BookSet (BS) is a (listof Book)
```

Here are a few examples of a **Book**:

```
(define weir (make-book "Weir" "The Martian" 369))
(define euclid (make-book "Euclid" "Elements" 654))
```

We can then make a **BS**, which can store a lot of information. For example:

```
(define booklist
  (list
    (make-book "Liu" "The Three Body Problem" 302)
    (make-book "Nawaz" "Songs for the End of the World" 400)
    (make-book "Heinlein" "The Moon Is a Harsh Mistress" 382)
    (make-book "Weir" "The Martian" 369)
    (make-book "Clancy" "The Sum of All Fears" 798)
    (make-book "Nawaz" "Bone and Bread" 445)
    (make-book "Heinlein" "Stranger in a Strange Land" 408)
    (make-book "Heinlein" "Starship Troopers" 263)))
```

Write a function `(total-pages books author)`, where `books` is a **BS**, and `author` is either a `Str` or `'any`. When `author` is a `Str`, the function produces the total number of pages in `books` written by that author. When `author` is `'any`, the function produces the total number of pages in `books` written by any author. For example:

```
(check-expect (total-pages booklist "Nawaz") 845)
(check-expect (total-pages booklist "Seuss") 0)
(check-expect (total-pages booklist 'any) 3367)
```

Copy the data definitions and convenience functions above into your solution, and use them. You do not need to provide design recipes for the provided constructor or selectors.

Submit your solution in the file `a05q3.rkt`.

Discussion

Topic: Assignments / Assignment 5

Hide Discussion

Add a Post

Show all posts | by recent activity

? Q1

By any chance, does it matter how racket presents the list of prime factors? My code returns stuff like (cons 2 (cons 3 '())) and I am wondering if that suffices

2